Oracle® Database

Reference

11*g* Release 1 (11.1)

B28320-01

July 2007

Oracle Database Reference, 11*g* Release 1 (11.1)

B28320-01

Primary Author: Tony Morales

Contributing Author: Kathy Rich

Contributors: Nipun Agarwal, Rick Anderson, Mark Bauer, Neerja Bhatt, Tudor Bosman, George Candea, Wilson Chan, Sumanta Chatterjee, Eugene Chong, Connie Dialeris Green, Harvey Eneman, Bruce Ernst, Ira Greenberg, Steve Harris, Thuvan Hoang, Lilian Hobbs, Namit Jain, Hakkan Jakobsson, Bob Jenkins, Mark Johnson, Jonathan Klein, Sushil Kumar, Tirthankar Lahiri, Bill Lee, Yunrui Li, Juan Loaiza, Rich Long, Diana Lorentz, Catherine Luu, Neil MacNaughton, Ari Mozes, Gopal Mulagund, Subramanian Muralidhar, Ravi Murthy, Karuna Muthiah, Arvind Nithrakashyap, Ronald Obermarck, Kant Patel, Christopher Racicot, Mark Ramacher, Shankar Raman, Suvendu Ray, Ann Rhee, Vivian Schupmann, Debbie Steiner, Seema Sundara, Juan Tellez, Alex Tsukerman, Kothanda Umamageswaran, Randy Urbano, Steve Vivian, Eric Voss, Tak Wang, Steve Wertheimer, Andy Witkowski, Daniel Wong, Graham Wood, Brian Wright, Aravind Yalamanchi, Qin Yu, Mohamed Ziauddin

Contents

3 Static Data Dictionary Views: ALL_PART_COL_STATISTICS to DATABASE_PROPERTIES

Part III Dynamic Performance Views

6 Dynamic Performance (V$) Views: V$ACCESS to V$HVMASTER_INFO

8 Dynamic Performance (V$) Views: V$SCHEDULER_RUNNING_JOBS to V$XS_SESSION_ROLE

A Database Limits

B SQL Scripts

C Oracle Wait Events

D Oracle Enqueue Names

E Statistics Descriptions

Index

Preface

This manual provides reference information about database initialization parameters, static data dictionary views, dynamic performance views, database limits, and SQL scripts that are part of the Oracle Database.

Oracle Database Reference contains information that describes the features and functionality of the Oracle Database (also known as the standard edition) and the Oracle Database Enterprise Edition products. The Oracle Database and the Oracle Database Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are optional. For example, to use application failover, you must have the Enterprise Edition with the Real Application Clusters option.

> **See Also:** *Oracle Database New Features Guide* for information about the differences between the Oracle Database and the Oracle Database Enterprise Edition and the features and options that are available to you.

This preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

Oracle Database Reference is intended for database administrators, system administrators, and database application developers.

To use this document, you need TO BE FAMILIAR WITH THE FOLLOWING:

- Oracle database management system (DBMS) concepts
- Your operating system environment

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see these Oracle resources:

- *Oracle Database Concepts* for a comprehensive introduction to the concepts and terminology used in this manual
- *Oracle Database Administrator's Guide* for information about administering the Oracle Database
- *Oracle Database Upgrade Guide* for the procedures for upgrading a previous release of Oracle to the new release
- *Oracle Database SQL Language Reference* for information on Oracle's SQL commands and functions
- *Oracle Database Advanced Application Developer's Guide* for information about developing database applications within the Oracle Database

Many of the examples in this book use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://www.oracle.com/technology`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at `http://www.oracle.com/technology`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in Oracle Database Reference?

This section describes new features of Oracle Database 11*g* release 1 (11.1) and provides pointers to additional information. For information on features that were new in earlier releases of the Oracle Database, refer to the documentation for the earlier release.

The following sections describe the new features in Oracle Database Reference:

- Oracle Database 11g Release 1 (11.1) New Features

Oracle Database 11g Release 1 (11.1) New Features

The following initialization parameters are new to release 11.1:

```
ASM_PREFERRED_READ_FAILURE_GROUPS
CLIENT_RESULT_CACHE_LAG
CLIENT_RESULT_CACHE_SIZE
COMMIT_LOGGING
COMMIT_WAIT
CONTROL_MANAGEMENT_PACK_ACCESS
DB_LOST_WRITE_PROTECT
DB_SECUREFILE
DB_ULTRA_SAFE
DDL_LOCK_TIMEOUT
DIAGNOSTIC_DEST
GLOBAL_TXN_PROCESSES
JAVA_JIT_ENABLED
LDAP_DIRECTORY_SYSAUTH
MEMORY_MAX_TARGET
MEMORY_TARGET
OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES
OPTIMIZER_USE_INVISIBLE_INDEXES
OPTIMIZER_USE_PENDING_STATISTICS
OPTIMIZER_USE_SQL_PLAN_BASELINES
PARALLEL_IO_CAP_ENABLED
PLSCOPE_SETTINGS
REDO_TRANSPORT_USER
RESOURCE_MANAGER_CPU_ALLOCATION
RESULT_CACHE_MAX_RESULT
RESULT_CACHE_MAX_SIZE
RESULT_CACHE_MODE
RESULT_CACHE_REMOTE_EXPIRATION
SEC_CASE_SENSITIVE_LOGON
```

```
SEC_MAX_FAILED_LOGIN_ATTEMPTS
SEC_PROTOCOL_ERROR_FURTHER_ACTION
SEC_PROTOCOL_ERROR_TRACE_ACTION
SEC_RETURN_SERVER_RELEASE_BANNER
XML_DB_EVENTS
```

> **See Also:** Part I, "Initialization Parameters" for more information about these initialization parameters

The following static data dictionary views are new to release 11.1:

ALL_ Views	DBA_ Views	USER_ Views
	DBA_ADDM_FDG_BREAKDOWN	USER_ADDM_FDG_BREAKDOWN
	DBA_ADDM_FINDINGS	USER_ADDM_FINDINGS
	DBA_ADDM_INSTANCES	USER_ADDM_INSTANCES
	DBA_ADDM_SYSTEM_DIRECTIVES	
	DBA_ADDM_TASK_DIRECTIVES	USER_ADDM_TASK_DIRECTIVES
	DBA_ADDM_TASKS	USER_ADDM_TASKS
	DBA_ADVISOR_DIR_DEFINITIONS	
	DBA_ADVISOR_DIR_INSTANCES	
	DBA_ADVISOR_DIR_TASK_INST	USER_ADVISOR_DIR_TASK_INST
	DBA_ADVISOR_EXEC_PARAMETERS	USER_ADVISOR_EXEC_PARAMETERS
	DBA_ADVISOR_EXECUTION_TYPES	
	DBA_ADVISOR_EXECUTIONS	USER_ADVISOR_EXECUTIONS
	DBA_ADVISOR_FDG_BREAKDOWN	USER_ADVISOR_FDG_BREAKDOWN
	DBA_ADVISOR_FINDING_NAMES	
	DBA_ADVISOR_SQLA_COLVOL	USER_ADVISOR_SQLA_COLVOL
	DBA_ADVISOR_SQLA_TABLES	USER_ADVISOR_SQLA_TABLES
	DBA_ADVISOR_SQLA_TABVOL	USER_ADVISOR_SQLA_TABVOL
	DBA_ADVISOR_SQLA_WK_SUM	USER_ADVISOR_SQLA_WK_SUM
	DBA_ADVISOR_SQLPLANS	USER_ADVISOR_SQLPLANS
	DBA_ADVISOR_SQLSTATS	USER_ADVISOR_SQLSTATS
	DBA_ARGUMENTS	
ALL_ASSEMBLIES	DBA_ASSEMBLIES	USER_ASSEMBLIES
	DBA_AUTOTASK_CLIENT	
	DBA_AUTOTASK_CLIENT_HISTORY	
	DBA_AUTOTASK_CLIENT_JOB	
	DBA_AUTOTASK_OPERATION	
	DBA_AUTOTASK_SCHEDULE	
	DBA_AUTOTASK_TASK	
	DBA_AUTOTASK_WINDOW_CLIENTS	
	DBA_AUTOTASK_WINDOW_HISTORY	
ALL_CHANGE_PROPAGATION_SETS		
ALL_CHANGE_PROPAGATIONS		
ALL_CHANGE_SETS		
ALL_CHANGE_SOURCES		

ALL_ Views	DBA_ Views	USER_ Views
ALL_CHANGE_TABLES		
ALL_COL_PENDING_STATS	DBA_COL_PENDING_STATS	USER_COL_PENDING_STATS
	DBA_COMPARISON	USER_COMPARISON
	DBA_COMPARISON_COLUMNS	USER_COMPARISON_COLUMNS
	DBA_COMPARISON_ROW_DIF	USER_COMPARISON_ROW_DIF
	DBA_COMPARISON_SCAN	USER_COMPARISON_SCAN
	DBA_COMPARISON_SCAN_SUMMARY	USER_COMPARISON_SCAN_SUMMARY
	DBA_COMPARISON_SCAN_VALUES	USER_COMPARISON_SCAN_VALUES
	DBA_CPOOL_INFO	
	DBA_CQ_NOTIFICATION_QUERIES	USER_CQ_NOTIFICATION_QUERIES
ALL_CUBE_ATTR_VISIBILITY	DBA_CUBE_ATTR_VISIBILITY	USER_CUBE_ATTR_VISIBILITY
ALL_CUBE_ATTRIBUTES	DBA_CUBE_ATTRIBUTES	USER_CUBE_ATTRIBUTES
ALL_CUBE_BUILD_PROCESSES	DBA_CUBE_BUILD_PROCESSES	USER_CUBE_BUILD_PROCESSES
ALL_CUBE_CALCULATED_MEMBERS	DBA_CUBE_CALCULATED_MEMBERS	USER_CUBE_CALCULATED_MEMBERS
ALL_CUBE_DIM_LEVELS	DBA_CUBE_DIM_LEVELS	USER_CUBE_DIM_LEVELS
ALL_CUBE_DIM_MODELS	DBA_CUBE_DIM_MODELS	USER_CUBE_DIM_MODELS
ALL_CUBE_DIM_VIEW_COLUMNS	DBA_CUBE_DIM_VIEW_COLUMNS	USER_CUBE_DIM_VIEW_COLUMNS
ALL_CUBE_DIM_VIEWS	DBA_CUBE_DIM_VIEWS	USER_CUBE_DIM_VIEWS
ALL_CUBE_DIMENSIONALITY	DBA_CUBE_DIMENSIONALITY	USER_CUBE_DIMENSIONALITY
ALL_CUBE_DIMENSIONS	DBA_CUBE_DIMENSIONS	USER_CUBE_DIMENSIONS
ALL_CUBE_HIER_LEVELS	DBA_CUBE_HIER_LEVELS	USER_CUBE_HIER_LEVELS
ALL_CUBE_HIER_VIEW_COLUMNS	DBA_CUBE_HIER_VIEW_COLUMNS	USER_CUBE_HIER_VIEW_COLUMNS
ALL_CUBE_HIER_VIEWS	DBA_CUBE_HIER_VIEWS	USER_CUBE_HIER_VIEWS
ALL_CUBE_HIERARCHIES	DBA_CUBE_HIERARCHIES	USER_CUBE_HIERARCHIES
ALL_CUBE_MEASURES	DBA_CUBE_MEASURES	USER_CUBE_MEASURES
ALL_CUBE_VIEW_COLUMNS	DBA_CUBE_VIEW_COLUMNS	USER_CUBE_VIEW_COLUMNS
ALL_CUBE_VIEWS	DBA_CUBE_VIEWS	USER_CUBE_VIEWS
ALL_CUBES	DBA_CUBES	USER_CUBES
	DBA_EPG_DAD_AUTHORIZATION	USER_EPG_DAD_AUTHORIZATION
	DBA_FLASHBACK_ARCHIVE	USER_FLASHBACK_ARCHIVE
	DBA_FLASHBACK_ARCHIVE_TABLES	USER_FLASHBACK_ARCHIVE_TABLES
	DBA_FLASHBACK_ARCHIVE_TS	
	DBA_FLASHBACK_TXN_REPORT	USER_FLASHBACK_TXN_REPORT
	DBA_FLASHBACK_TXN_STATE	USER_FLASHBACK_TXN_STATE
	DBA_HIST_BASELINE_DETAILS	
	DBA_HIST_BASELINE_METADATA	
	DBA_HIST_BASELINE_TEMPLATE	
	DBA_HIST_CLUSTER_INTERCON	
	DBA_HIST_COLORED_SQL	
	DBA_HIST_EVENT_HISTOGRAM	
	DBA_HIST_IC_CLIENT_STATS	
	DBA_HIST_IC_DEVICE_STATS	
	DBA_HIST_INTERCONNECT_PINGS	

ALL_ Views	DBA_ Views	USER_ Views
	DBA_HIST_IOSTAT_FILETYPE	
	DBA_HIST_IOSTAT_FILETYPE_NAME	
	DBA_HIST_IOSTAT_FUNCTION	
	DBA_HIST_IOSTAT_FUNCTION_NAME	
	DBA_HIST_MEM_DYNAMIC_COMP	
	DBA_HIST_MEMORY_RESIZE_OPS	
	DBA_HIST_MEMORY_TARGET_ADVICE	
	DBA_HIST_MUTEX_SLEEP	
	DBA_HIST_PERSISTENT_QUEUES	
	DBA_HIST_PERSISTENT_SUBS	
	DBA_HIST_RSRC_CONSUMER_GROUP	
	DBA_HIST_RSRC_PLAN	
ALL_IDENTIFIERS	DBA_IDENTIFIERS	USER_IDENTIFIERS
ALL_IND_PENDING_STATS	DBA_IND_PENDING_STATS	USER_IND_PENDING_STATS
	DBA_INVALID_OBJECTS	
ALL_JAVA_COMPILER_OPTIONS	DBA_JAVA_COMPILER_OPTIONS	USER_JAVA_COMPILER_OPTIONS
	DBA_LOGSTDBY_UNSUPPORTED_TABLE	
ALL_MEASURE_FOLDER_CONTENTS	DBA_MEASURE_FOLDER_CONTENTS	USER_MEASURE_FOLDER_CONTENTS
ALL_MEASURE_FOLDERS	DBA_MEASURE_FOLDERS	USER_MEASURE_FOLDERS
ALL_MINING_MODEL_ATTRIBUTES	DBA_MINING_MODEL_ATTRIBUTES	USER_MINING_MODEL_ATTRIBUTES
ALL_MINING_MODEL_SETTINGS	DBA_MINING_MODEL_SETTINGS	USER_MINING_MODEL_SETTINGS
ALL_MINING_MODELS	DBA_MINING_MODELS	USER_MINING_MODELS
ALL_MVIEW_DETAIL_PARTITION	DBA_MVIEW_DETAIL_PARTITION	USER_MVIEW_DETAIL_PARTITION
ALL_MVIEW_DETAIL_SUBPARTITION	DBA_MVIEW_DETAIL_SUBPARTITION	USER_MVIEW_DETAIL_SUBPARTITION
	DBA_NETWORK_ACL_PRIVILEGES	USER_NETWORK_ACL_PRIVILEGES
	DBA_NETWORK_ACLS	
	DBA_OLDIMAGE_COLUMNS	USER_OLDIMAGE_COLUMNS
	DBA_REGISTRY_DATABASE	
	DBA_REGISTRY_DEPENDENCIES	
	DBA_REGISTRY_PROGRESS	
	DBA_RSRC_CAPABILITY	
	DBA_RSRC_CATEGORIES	
	DBA_RSRC_INSTANCE_CAPABILITY	
	DBA_RSRC_IO_CALIBRATE	
	DBA_RSRC_STORAGE_POOL_MAPPING	
ALL_SCHEDULER_CREDENTIALS	DBA_SCHEDULER_CREDENTIALS	USER_SCHEDULER_CREDENTIALS
	DBA_SCHEDULER_JOB_ROLES	
ALL_SCHEDULER_REMOTE_DATABASES	DBA_SCHEDULER_REMOTE_DATABASES	
ALL_SCHEDULER_REMOTE_JOBSTATE	DBA_SCHEDULER_REMOTE_JOBSTATE	USER_SCHEDULER_REMOTE_JOBSTATE
	DBA_SQL_MANAGEMENT_CONFIG	
	DBA_SQL_PATCHES	
	DBA_SQL_PLAN_BASELINES	
ALL_STAT_EXTENSIONS	DBA_STAT_EXTENSIONS	USER_STAT_EXTENSIONS

ALL_ Views	DBA_ Views	USER_ Views
ALL_STREAMS_COLUMNS	DBA_STREAMS_COLUMNS	
	DBA_STREAMS_TP_COMPONENT	
	DBA_STREAMS_TP_COMPONENT_LINK	
	DBA_STREAMS_TP_COMPONENT_STAT	
	DBA_STREAMS_TP_DATABASE	
	DBA_STREAMS_TP_PATH_BOTTLENECK	
	DBA_STREAMS_TP_PATH_STAT	
	DBA_SUBSCR_REGISTRATIONS	USER_SUBSCR_REGISTRATIONS
ALL_SYNC_CAPTURE	DBA_SYNC_CAPTURE	
ALL_SYNC_CAPTURE_PREPARED_TABS	DBA_SYNC_CAPTURE_PREPARED_TABS	
ALL_SYNC_CAPTURE_TABLES	DBA_SYNC_CAPTURE_TABLES	
ALL_TAB_HISTGRM_PENDING_STATS	DBA_TAB_HISTGRM_PENDING_STATS	USER_TAB_HISTGRM_PENDING_STATS
ALL_TAB_PENDING_STATS	DBA_TAB_PENDING_STATS	USER_TAB_PENDING_STATS
ALL_TAB_STAT_PREFS	DBA_TAB_STAT_PREFS	USER_TAB_STAT_PREFS
	DBA_TEMP_FREE_SPACE	
ALL_TRIGGER_ORDERING	DBA_TRIGGER_ORDERING	USER_TRIGGER_ORDERING
	DBA_USERS_WITH_DEFPWD	
	DBA_WORKLOAD_CAPTURES	
	DBA_WORKLOAD_CONNECTION_MAP	
	DBA_WORKLOAD_FILTERS	
	DBA_WORKLOAD_REPLAY_DIVERGENCE	
	DBA_WORKLOAD_REPLAYS	

See Also: Part II, "Static Data Dictionary Views" for more information about these static data dictionary views

The following dynamic performance views are new to release 11.1:

GV$ Views	V$ Views
GV$ASM_ATTRIBUTE	V$ASM_ATTRIBUTE
GV$ASM_DISK_IOSTAT	V$ASM_DISK_IOSTAT
GV$CALLTAG	V$CALLTAG
GV$CLIENT_RESULT_CACHE_STATS	V$CLIENT_RESULT_CACHE_STATS
GV$CORRUPT_XID_LIST	V$CORRUPT_XID_LIST
GV$CPOOL_CC_INFO	V$CPOOL_CC_INFO
GV$CPOOL_CC_STATS	V$CPOOL_CC_STATS
GV$CPOOL_STATS	V$CPOOL_STATS
GV$DETACHED_SESSION	V$DETACHED_SESSION
GV$DIAG_INFO	V$DIAG_INFO
GV$DNFS_CHANNELS	V$DNFS_CHANNELS
GV$DNFS_FILES	V$DNFS_FILES
GV$DNFS_SERVERS	V$DNFS_SERVERS

GV$ Views	V$ Views
GV$DNFS_STATS	V$DNFS_STATS
GV$DYNAMIC_REMASTER_STATS	V$DYNAMIC_REMASTER_STATS
GV$ENCRYPTED_TABLESPACES	V$ENCRYPTED_TABLESPACES
GV$ENCRYPTION_WALLET	V$ENCRYPTION_WALLET
	V$FLASHBACK_TXN_GRAPH
	V$FLASHBACK_TXN_MODS
GV$FOREIGN_ARCHIVED_LOG	V$FOREIGN_ARCHIVED_LOG
GV$FS_FAILOVER_HISTOGRAM	V$FS_FAILOVER_HISTOGRAM
GV$FS_FAILOVER_STATS	V$FS_FAILOVER_STATS
GV$HM_CHECK	V$HM_CHECK
GV$HM_CHECK_PARAM	V$HM_CHECK_PARAM
GV$HM_FINDING	V$HM_FINDING
GV$HM_INFO	V$HM_INFO
GV$HM_RECOMMENDATION	V$HM_RECOMMENDATION
GV$HM_RUN	V$HM_RUN
GV$INCMETER_CONFIG	V$INCMETER_CONFIG
GV$INCMETER_INFO	V$INCMETER_INFO
GV$INCMETER_SUMMARY	V$INCMETER_SUMMARY
GV$IOFUNCMETRIC	V$IOFUNCMETRIC
GV$IOFUNCMETRIC_HISTORY	V$IOFUNCMETRIC_HISTORY
GV$IOSTAT_CONSUMER_GROUP	V$IOSTAT_CONSUMER_GROUP
GV$IOSTAT_FILE	V$IOSTAT_FILE
GV$IOSTAT_FUNCTION	V$IOSTAT_FUNCTION
GV$IOSTAT_NETWORK	V$IOSTAT_NETWORK
GV$IO_CALIBRATION_STATUS	V$IO_CALIBRATION_STATUS
GV$IR_FAILURE	V$IR_FAILURE
GV$IR_FAILURE_SET	V$IR_FAILURE_SET
GV$IR_MANUAL_CHECKLIST	V$IR_MANUAL_CHECKLIST
GV$IR_REPAIR	V$IR_REPAIR
GV$LOBSTAT	V$LOBSTAT
GV$MEMORY_CURRENT_RESIZE_OPS	V$MEMORY_CURRENT_RESIZE_OPS
GV$MEMORY_DYNAMIC_COMPONENTS	V$MEMORY_DYNAMIC_COMPONENTS
GV$MEMORY_RESIZE_OPS	V$MEMORY_RESIZE_OPS
GV$MEMORY_TARGET_ADVICE	V$MEMORY_TARGET_ADVICE
GV$NFS_CLIENTS	V$NFS_CLIENTS
GV$NFS_LOCKS	V$NFS_LOCKS
GV$NFS_OPEN_FILES	V$NFS_OPEN_FILES
	V$OBJECT_PRIVILEGE

GV$ Views	V$ Views
GV$PERSISTENT_PUBLISHERS	V$PERSISTENT_PUBLISHERS
GV$PERSISTENT_QUEUES	V$PERSISTENT_QUEUES
GV$PERSISTENT_SUBSCRIBERS	V$PERSISTENT_SUBSCRIBERS
GV$PROCESS_GROUP	V$PROCESS_GROUP
GV$PX_INSTANCE_GROUP	V$PX_INSTANCE_GROUP
GV$REDO_DEST_RESP_HISTOGRAM	V$REDO_DEST_RESP_HISTOGRAM
GV$RESULT_CACHE_DEPENDENCY	V$RESULT_CACHE_DEPENDENCY
GV$RESULT_CACHE_MEMORY	V$RESULT_CACHE_MEMORY
GV$RESULT_CACHE_OBJECTS	V$RESULT_CACHE_OBJECTS
GV$RESULT_CACHE_STATISTICS	V$RESULT_CACHE_STATISTICS
GV$RMAN_COMPRESSION_ALGORITHM	V$RMAN_COMPRESSION_ALGORITHM
	V$RMAN_ENCRYPTION_ALGORITHMS
GV$RSRCMGRMETRIC	V$RSRCMGRMETRIC
GV$RSRCMGRMETRIC_HISTORY	V$RSRCMGRMETRIC_HISTORY
GV$SECUREFILE_TIMER	V$SECUREFILE_TIMER
	V$SESSION_FIX_CONTROL
GV$SQL_CS_HISTOGRAM	V$SQL_CS_HISTOGRAM
GV$SQL_CS_SELECTIVITY	V$SQL_CS_SELECTIVITY
GV$SQL_CS_STATISTICS	V$SQL_CS_STATISTICS
GV$SQL_FEATURE	V$SQL_FEATURE
GV$SQL_FEATURE_DEPENDENCY	V$SQL_FEATURE_DEPENDENCY
GV$SQL_FEATURE_HIERARCHY	V$SQL_FEATURE_HIERARCHY
GV$SQL_HINT	V$SQL_HINT
GV$SQL_MONITOR	V$SQL_MONITOR
GV$SQL_PLAN_MONITOR	V$SQL_PLAN_MONITOR
GV$SQLFN_ARG_METADATA	V$SQLFN_ARG_METADATA
GV$SQLFN_METADATA	V$SQLFN_METADATA
GV$SSCR_SESSIONS	V$SSCR_SESSIONS
GV$STREAMS_MESSAGE_TRACKING	V$STREAMS_MESSAGE_TRACKING
GV$SUBSCR_REGISTRATION_STATS	V$SUBSCR_REGISTRATION_STATS
	V$SYSTEM_FIX_CONTROL
	V$WAIT_CHAINS
GV$WORKLOAD_REPLAY_THREAD	V$WORKLOAD_REPLAY_THREAD
	V$XS_SESSION
	V$XS_SESSION_ATTRIBUTE
	V$XS_SESSION_ROLE

See Also: Part III, "Dynamic Performance Views" for more information about these dynamic performance views

Part I

Initialization Parameters

This part describes the database initialization parameters that can be specified in a parameter file to start or configure an instance. This part contains the following chapter:

- Chapter 1, "Initialization Parameters"

1

Initialization Parameters

This chapter contains detailed descriptions (in alphabetical order) of the database initialization parameters.

This chapter contains the following topics:

Uses of Initialization Parameters

Initialization parameters fall into various functional groups. For example, parameters perform the following functions:

- Set limits for the entire database
- Set user or process limits
- Set limits on database resources
- Affect performance (these are called **variable parameters**)

  Variable parameters are of particular interest to database administrators, because these parameters are used primarily to improve database performance.

Database administrators can use initialization parameters to:

- Optimize performance by adjusting memory structures, such as the number of database buffers in memory
- Set database-wide defaults, such as the amount of space initially allocated for a context area when it is created
- Set database limits, such as the maximum number of database users
- Specify names of files or directories required by the database

Many initialization parameters can be fine-tuned to improve database performance. Other parameters should never be altered or should be altered only under the supervision of Oracle Support Services.

All initialization parameters are optional. Oracle has a default value for each parameter. This value may be operating system-dependent, depending on the parameter.

Types of Initialization Parameters

The Oracle database server has the following types of initialization parameters:

- Derived Parameters
- Operating System-Dependent Parameters
- Variable Parameters (these can be dynamic parameters or any of the preceding ones)

Derived Parameters

Some initialization parameters are **derived**, meaning that their values are calculated from the values of other parameters. Normally, you should not alter values for derived parameters, but if you do, then the value you specify will override the calculated value.

For example, the default value of the `SESSIONS` parameter is derived from the value of the `PROCESSES` parameter. If the value of `PROCESSES` changes, then the default value of `SESSIONS` changes as well, unless you override it with a specified value.

Operating System-Dependent Parameters

The valid values or value ranges of some initialization parameters depend upon the host operating system. For example, the parameter `DB_BLOCK_BUFFERS` indicates the number of data buffers in main memory, and its maximum value depends on the operating system. The size of those buffers, set by `DB_BLOCK_SIZE`, has an operating system-dependent default value.

> **See Also:** Your operating system-specific Oracle documentation for more information on operating system-dependent Oracle parameters and operating system parameters

Variable Parameters

The variable initialization parameters offer the most potential for improving system performance. Some variable parameters set capacity limits but do not affect performance. For example, when the value of `OPEN_CURSORS` is 10, a user process attempting to open its eleventh cursor receives an error. Other variable parameters affect performance but do not impose absolute limits. For example, reducing the value of `DB_BLOCK_BUFFERS` does not prevent work even though it may slow down performance.

Increasing the values of variable parameters may improve your system's performance, but increasing most parameters also increases the system global area (SGA) size. A larger SGA can improve database performance up to a point. In virtual memory operating systems, an SGA that is too large can degrade performance if it is swapped in and out of memory. Operating system parameters that control virtual memory working areas should be set with the SGA size in mind. The operating system configuration can also limit the maximum size of the SGA.

Basic Initialization Parameters

The following is a list of the database basic initialization parameters. Most databases should only need to have basic parameters set to run properly and efficiently. Oracle advises you to become familiar with the basic parameters and only use other parameters when directed to by feature documentation or in special circumstances:

```
CLUSTER_DATABASE
COMPATIBLE
CONTROL_FILES
DB_BLOCK_SIZE
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_DOMAIN
DB_NAME
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_UNIQUE_NAME
INSTANCE_NUMBER
LDAP_DIRECTORY_SYSAUTH
LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n
NLS_LANGUAGE
NLS_TERRITORY
OPEN_CURSORS
PGA_AGGREGATE_TARGET
PROCESSES
REMOTE_LISTENER
REMOTE_LOGIN_PASSWORDFILE
ROLLBACK_SEGMENTS
SESSIONS
SGA_TARGET
SHARED_SERVERS
STAR_TRANSFORMATION_ENABLED
UNDO_MANAGEMENT
UNDO_TABLESPACE
```

Parameter Files

A **parameter file** is a file that contains a list of initialization parameters and a value for each parameter. You specify initialization parameters in a parameter file that reflect your particular installation. Oracle supports the following two types of parameter files:

- Server Parameter Files
- Initialization Parameter Files

Server Parameter Files

A **server parameter file** is a binary file that acts as a repository for initialization parameters. The server parameter file can reside on the machine where the Oracle database server executes. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.

Initialization Parameter Files

An **initialization parameter file** is a text file that contains a list of initialization parameters. The file should be written in the client's default character set.

The following are sample entries in an initialization parameter file:

```
PROCESSES = 100
OPEN_LINKS = 12
GLOBAL_NAMES = true
```

The name of the initialization parameter file varies depending on the operating system. For example, it can be in mixed case or lowercase, or it can have a logical name or a variation of the name `init.ora`. Also supplied is an `initdw.ora` file, which contains suggested parameter settings for data warehouses and data marts. The database administrator can choose a different filename for the initialization parameter file.

Refer to your operating system-specific Oracle documentation for the default locations and filenames for initialization parameter files on your operating system. The initialization parameter file is read by the client-side tool used to start the server (such as SQL*Plus).

Sample initialization parameter files are provided on the Oracle distribution medium for each operating system. A sample file is sufficient for initial use, but you will probably want to modify the file to tune the database for best performance. Any changes will take effect after you completely shut down and restart the instance.

Specifying Values in an Initialization Parameter File

This section describes several aspects of setting parameter values in an initialization parameter file.

Rules Governing Initialization Parameter Files The following rules govern the specification of parameters in initialization parameter files:

- An initialization parameter file should contain only parameters and comments. A pound sign (#) starts a comment line. The rest of the line is ignored.
- You can specify parameters in any order.
- Case (upper or lower) in filenames is significant only if case is significant on the host operating system.
- To enter several parameters on one line, use spaces between parameter names and values, as in the following example:

  ```
  PROCESSES = 100 CPU_COUNT = 1 OPEN_CURSORS = 10
  ```

- Some parameters, such as `ROLLBACK_SEGMENTS`, accept multiple value entries. Enter multiple values enclosed in parentheses and separated by commas. For example:

  ```
  ROLLBACK_SEGMENTS = (SEG1, SEG2, SEG3, SEG4, SEG5)
  ```

  Alternatively, you can enter multiple values without parentheses and commas. For example:

  ```
  ROLLBACK_SEGMENTS = SEG1 SEG2 SEG3 SEG4 SEG5
  ```

  Either syntax is valid.

If you enter values for one parameter in multiple entries, then the entries must be on consecutive lines. If they are not, then the first entry will not be processed properly. For example, in the following entry the setting for `SEG3` and `SEG4` will override the setting for `SEG1` and `SEG2`:

```
ROLLBACK_SEGMENTS = SEG1 SEG2
OPEN_CURSORS = 10
ROLLBACK_SEGMENTS = SEG3 SEG4
```

- A backslash (\), also known as an escape character, indicates continuation of the parameter specification. If a backslash continues a line, then the continued line must have no leading spaces. For example:

```
ROLLBACK_SEGMENTS = (SEG1, SEG2, \
SEG3, SEG4, SEG5)
```

- You can use the `IFILE` initialization parameter to embed the contents of another initialization parameter file into the current initialization parameter file.
- Enclose in quotation marks any parameter values that contain spaces or tabs. You can use either single or double quotation marks unless otherwise indicated. For example:

```
NLS_TERRITORY = 'CZECH REPUBLIC'
```

> **Note:** Listing parameters in alphabetical order in the initialization parameter file can help you to find them and can help ensure that each parameter is specified only once.

- Enclose in quotation marks any parameter value that contains a special character.

See Also:

- Your operating system-specific Oracle documentation for more information on initialization parameter files
- "IFILE" on page 1-69

Using Special Characters in Parameter Values If a parameter value contains a special character, then the special character must be preceded by a backslash or the entire parameter value must be enclosed in quotation marks. For example, you can specify special characters using either of the following:

```
DB_DOMAIN = 'JAPAN.ACME#.COM'

DB_DOMAIN = JAPAN.ACME\#.COM
```

Table 1–1 lists the special characters you can use in initialization parameter files.

Table 1–1 Special Characters in Initialization Parameter Files

Character	Name	Description
#	Number sign	Comment
(	Left parenthesis	Start list of values
)	Right parenthesis	End list of values

Table 1–1 (Cont.) Special Characters in Initialization Parameter Files

Character	Name	Description
"	Double quotation mark	Start or end of quoted string
'	Single quotation mark	Start or end of quoted string
=	Equal sign	Separator of keyword and values
,	Comma	Separator of elements
-	Minus sign	Precedes UNIX-style keywords
\	Backslash	Escape character

If a special character must be treated literally in an initialization parameter file, then it must either be preceded by the backslash character, or the entire string containing the special character must be enclosed in quotation marks.

Using the Escape Character As described in "Rules Governing Initialization Parameter Files" on page 1-4, the backslash (\) can also signify a line continuation. If the backslash is followed by an alphanumeric character, then the backslash is treated as a normal character in the input. If it is not followed by an alphanumeric character, then the backslash is treated either as a backslash or as a continuation character.

Using Quotation Marks Quotation marks can be nested in one of two ways. The first method is to double the quotation marks in the nested string. For example:

```
NLS_DATE_FORMAT = '''Today is'' MM/DD/YYYY'
```

The second method is to alternate single and double quotation marks. For example:

```
NLS_DATE_FORMAT = '"Today is" MM/DD/YYYY'
```

Changing Parameter Values in a Parameter File

You change the value of a parameter in a parameter file in one of the following ways:

- By editing an initialization parameter file

  In most cases, the new value takes effect the next time you start an instance of the database.

- By issuing an `ALTER SYSTEM SET ... SCOPE=SPFILE` statement to update a server parameter file

Parameters by Functional Category

The following list shows the initialization parameters by their functional category:

- **ANSI Compliance**

  ```
  BLANK_TRIMMING
  ```

- **Backup and Restore**

  ```
  BACKUP_TAPE_IO_SLAVES
  RECYCLEBIN
  TAPE_ASYNCH_IO
  ```

- **BFILEs**

  ```
  SESSION_MAX_OPEN_FILES
  ```

- **Buffer Cache and I/O**

  ```
  CLIENT_RESULT_CACHE_LAG
  CLIENT_RESULT_CACHE_SIZE
  DB_nK_CACHE_SIZE
  DB_BLOCK_BUFFERS
  DB_BLOCK_SIZE
  DB_CACHE_ADVICE
  DB_CACHE_SIZE
  DB_FILE_MULTIBLOCK_READ_COUNT
  DB_KEEP_CACHE_SIZE
  DB_RECYCLE_CACHE_SIZE
  DB_WRITER_PROCESSES
  DBWR_IO_SLAVES
  DISK_ASYNCH_IO
  FILESYSTEMIO_OPTIONS
  READ_ONLY_OPEN_DELAYED
  RESULT_CACHE_MAX_RESULT
  RESULT_CACHE_MAX_SIZE
  RESULT_CACHE_MODE
  USE_INDIRECT_DATA_BUFFERS
  ```

- **Cursors and Library Cache**

  ```
  CURSOR_SHARING
  CURSOR_SPACE_FOR_TIME
  OPEN_CURSORS
  SESSION_CACHED_CURSORS
  ```

- **Database/Instance Identification**

  ```
  DB_DOMAIN
  DB_NAME
  INSTANCE_NAME
  ```

- **Diagnostics and Statistics**

  ```
  BACKGROUND_CORE_DUMP
  BACKGROUND_DUMP_DEST
  CORE_DUMP_DEST
  DB_BLOCK_CHECKING
  DB_BLOCK_CHECKSUM
  DIAGNOSTIC_DEST
  EVENT
  MAX_DUMP_FILE_SIZE
  SHADOW_CORE_DUMP
  STATISTICS_LEVEL
  TIMED_OS_STATISTICS
  TIMED_STATISTICS
  TRACE_ENABLED
  TRACEFILE_IDENTIFIER
  USER_DUMP_DEST
  ```

- **Distributed, Replication**

COMMIT_POINT_STRENGTH
DISTRIBUTED_LOCK_TIMEOUT
GLOBAL_NAMES
HS_AUTOREGISTER
OPEN_LINKS
OPEN_LINKS_PER_INSTANCE
REPLICATION_DEPENDENCY_TRACKING

- **File Locations, Names, and Sizes**

  AUDIT_FILE_DEST
  BACKGROUND_CORE_DUMP
  BACKGROUND_DUMP_DEST
  CONTROL_FILES
  CORE_DUMP_DEST
  DB_CREATE_FILE_DEST
  DB_CREATE_ONLINE_LOG_DEST_*n*
  DB_FILES
  DB_RECOVERY_FILE_DEST
  DB_RECOVERY_FILE_DEST_SIZE
  FILE_MAPPING
  IFILE
  LOG_ARCHIVE_DEST_*n*
  SPFILE

- **Globalization**

  NLS_CALENDAR
  NLS_COMP
  NLS_CURRENCY
  NLS_DATE_FORMAT
  NLS_DATE_LANGUAGE
  NLS_DUAL_CURRENCY
  NLS_ISO_CURRENCY
  NLS_LANGUAGE
  NLS_LENGTH_SEMANTICS
  NLS_NCHAR_CONV_EXCP
  NLS_NUMERIC_CHARACTERS
  NLS_SORT
  NLS_TERRITORY
  NLS_TIMESTAMP_FORMAT
  NLS_TIMESTAMP_TZ_FORMAT

- **Java**

  JAVA_JIT_ENABLED
  JAVA_MAX_SESSIONSPACE_SIZE
  JAVA_POOL_SIZE
  JAVA_SOFT_SESSIONSPACE_LIMIT

- **Job Queues**

  JOB_QUEUE_PROCESSES

- **License Limits**

  LICENSE_MAX_SESSIONS
  LICENSE_MAX_USERS

```
LICENSE_SESSIONS_WARNING
```

- **Memory**

  ```
  MEMORY_MAX_TARGET
  MEMORY_TARGET
  ```

- **Miscellaneous**

  ```
  AQ_TM_PROCESSES
  ASM_PREFERRED_READ_FAILURE_GROUPS
  COMPATIBLE
  FIXED_DATE
  LDAP_DIRECTORY_SYSAUTH
  XML_DB_EVENTS
  ```

- **Networking**

  ```
  LOCAL_LISTENER
  REMOTE_LISTENER
  SERVICE_NAMES
  ```

- **Objects and LOBs**

  ```
  OBJECT_CACHE_MAX_SIZE_PERCENT
  OBJECT_CACHE_OPTIMAL_SIZE
  ```

- **OLAP**

  ```
  OLAP_PAGE_POOL_SIZE
  ```

- **Optimizer**

  ```
  OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES
  OPTIMIZER_DYNAMIC_SAMPLING
  OPTIMIZER_FEATURES_ENABLE
  OPTIMIZER_INDEX_CACHING
  OPTIMIZER_INDEX_COST_ADJ
  OPTIMIZER_MODE
  OPTIMIZER_SECURE_VIEW_MERGING
  OPTIMIZER_USE_PENDING_STATISTICS
  OPTIMIZER_USE_SQL_PLAN_BASELINES
  QUERY_REWRITE_ENABLED
  QUERY_REWRITE_INTEGRITY
  STAR_TRANSFORMATION_ENABLED
  ```

- **Parallel Execution**

  ```
  PARALLEL_ADAPTIVE_MULTI_USER
  PARALLEL_EXECUTION_MESSAGE_SIZE
  PARALLEL_MAX_SERVERS
  PARALLEL_MIN_PERCENT
  PARALLEL_MIN_SERVERS
  PARALLEL_THREADS_PER_CPU
  ```

- **PL/SQL**

  ```
  PLSQL_NATIVE_LIBRARY_DIR
  PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
  ```

PLSQL_V2_COMPATIBILITY
REMOTE_DEPENDENCIES_MODE
UTL_FILE_DIR

- **PL/SQL Compiler**

  PLSCOPE_SETTINGS
  PLSQL_CCFLAGS
  PLSQL_CODE_TYPE
  PLSQL_DEBUG
  PLSQL_OPTIMIZE_LEVEL
  PLSQL_WARNINGS
  NLS_LENGTH_SEMANTICS

- **SGA Memory**

  DB_*n*K_CACHE_SIZE
  DB_CACHE_SIZE
  HI_SHARED_MEMORY_ADDRESS
  JAVA_POOL_SIZE
  LARGE_POOL_SIZE
  LOCK_SGA
  OLAP_PAGE_POOL_SIZE
  PRE_PAGE_SGA
  SGA_MAX_SIZE
  SGA_TARGET
  SHARED_MEMORY_ADDRESS
  SHARED_POOL_RESERVED_SIZE
  SHARED_POOL_SIZE
  STREAMS_POOL_SIZE

- **Real Application Clusters**

  ACTIVE_INSTANCE_COUNT
  CLUSTER_DATABASE
  CLUSTER_DATABASE_INSTANCES
  CLUSTER_INTERCONNECTS
  GC_FILES_TO_LOCKS
  INSTANCE_NUMBER
  PARALLEL_INSTANCE_GROUP
  THREAD

- **Redo Logs, Archiving, and Recovery**

  CONTROL_FILE_RECORD_KEEP_TIME
  DB_CREATE_ONLINE_LOG_DEST_*n*
  DB_RECOVERY_FILE_DEST
  DB_RECOVERY_FILE_DEST_SIZE
  FAST_START_MTTR_TARGET
  LOG_BUFFER
  LOG_CHECKPOINT_INTERVAL
  LOG_CHECKPOINT_TIMEOUT
  LOG_CHECKPOINTS_TO_ALERT
  LOG_ARCHIVE_CONFIG
  LOG_ARCHIVE_DEST_*n*
  LOG_ARCHIVE_DEST_STATE_*n*
  LOG_ARCHIVE_DUPLEX_DEST

```
LOG_ARCHIVE_FORMAT
LOG_ARCHIVE_MAX_PROCESSES
LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_ARCHIVE_TRACE
REDO_TRANSPORT_USER
RECOVERY_PARALLELISM
```

- **Resource Manager**

```
RESOURCE_LIMIT
RESOURCE_MANAGER_CPU_ALLOCATION
RESOURCE_MANAGER_PLAN
```

- **Security and Auditing**

```
AUDIT_FILE_DEST
AUDIT_SYS_OPERATIONS
AUDIT_SYSLOG_LEVEL
AUDIT_TRAIL
COMMIT_LOGGING
COMMIT_WAIT
O7_DICTIONARY_ACCESSIBILITY
OS_AUTHENT_PREFIX
OS_ROLES
RDBMS_SERVER_DN
REMOTE_LOGIN_PASSWORDFILE
REMOTE_OS_AUTHENT
REMOTE_OS_ROLES
SEC_CASE_SENSITIVE_LOGON
SEC_MAX_FAILED_LOGIN_ATTEMPTS
SEC_PROTOCOL_ERROR_FURTHER_ACTION
SEC_PROTOCOL_ERROR_TRACE_ACTION
SEC_RETURN_SERVER_RELEASE_BANNER
SQL92_SECURITY
```

- **Sessions and Processes**

```
CPU_COUNT
PROCESSES
SESSIONS
```

- **Shared Server Architecture**

```
CIRCUITS
DISPATCHERS
MAX_DISPATCHERS
MAX_SHARED_SERVERS
SHARED_SERVER_SESSIONS
SHARED_SERVERS
```

- **Standby Database**

```
ARCHIVE_LAG_TARGET
DB_FILE_NAME_CONVERT
DB_UNIQUE_NAME
DG_BROKER_CONFIG_FILEn
DG_BROKER_START
FAL_CLIENT
```

```
FAL_SERVER
LOG_FILE_NAME_CONVERT
STANDBY_FILE_MANAGEMENT
```

- **Temporary Sort Space**

```
BITMAP_MERGE_AREA_SIZE
CREATE_BITMAP_AREA_SIZE
HASH_AREA_SIZE
PGA_AGGREGATE_TARGET
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
WORKAREA_SIZE_POLICY
```

- **Transactions**

```
COMMIT_LOGGING
COMMIT_WAIT
DB_LOST_WRITE_PROTECT
DDL_LOCK_TIMEOUT
DML_LOCKS
FAST_START_PARALLEL_ROLLBACK
GLOBAL_TXN_PROCESSES
TRANSACTIONS
```

- **Undo Management**

```
RESUMABLE_TIMEOUT
ROLLBACK_SEGMENTS
TRANSACTIONS_PER_ROLLBACK_SEGMENT
UNDO_MANAGEMENT
UNDO_RETENTION
UNDO_TABLESPACE
```

Modifiable Parameters

Some initialization parameters can be modified using the ALTER SESSION or ALTER SYSTEM statements while an instance is running. Use the following syntax to modify initialization parameters:

```
ALTER SESSION SET parameter_name = value
ALTER SYSTEM SET parameter_name = value [DEFERRED]
```

Whenever a parameter is modified using the ALTER SYSTEM statement, the Oracle Database records the statement that modifies the parameter in the alert log.

The ALTER SESSION statement changes the value of the specified parameter for the duration of the session that invokes the statement. The value of the parameter does not change for other sessions in the instance. The value of the following initialization parameters can be changed with ALTER SESSION:

```
ASM_POWER_LIMIT
COMMIT_LOGGING
COMMIT_WAIT
COMMIT_WRITE
CREATE_STORED_OUTLINES
CURSOR_SHARING
```

DB_BLOCK_CHECKING

DB_CREATE_FILE_DEST

DB_CREATE_ONLINE_LOG_DEST_*n*

DB_FILE_MULTIBLOCK_READ_COUNT

DB_FILE_NAME_CONVERT

DB_SECUREFILE

DDL_LOCK_TIMEOUT

GLOBAL_NAMES

HASH_AREA_SIZE

JAVA_JIT_ENABLED

LOG_ARCHIVE_DEST_*n*

LOG_ARCHIVE_DEST_STATE_*n*

LOG_ARCHIVE_MIN_SUCCEED_DEST

MAX_DUMP_FILE_SIZE

NLS_CALENDAR

NLS_COMP

NLS_CURRENCY

NLS_DATE_FORMAT

NLS_DATE_LANGUAGE

NLS_DUAL_CURRENCY

NLS_ISO_CURRENCY

NLS_LANGUAGE

NLS_LENGTH_SEMANTICS

NLS_NCHAR_CONV_EXCP

NLS_NUMERIC_CHARACTERS

NLS_SORT

NLS_TERRITORY

NLS_TIMESTAMP_FORMAT

NLS_TIMESTAMP_TZ_FORMAT

OBJECT_CACHE_MAX_SIZE_PERCENT

OBJECT_CACHE_OPTIMAL_SIZE

OLAP_PAGE_POOL_SIZE

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES

OPTIMIZER_DYNAMIC_SAMPLING

OPTIMIZER_FEATURES_ENABLE

OPTIMIZER_INDEX_CACHING

OPTIMIZER_INDEX_COST_ADJ

OPTIMIZER_MODE

OPTIMIZER_USE_INVISIBLE_INDEXES

OPTIMIZER_USE_PENDING_STATISTICS

OPTIMIZER_USE_SQL_PLAN_BASELINES

PARALLEL_INSTANCE_GROUP

PARALLEL_IO_CAP_ENABLED

PARALLEL_MIN_PERCENT

PLSCOPE_SETTINGS

PLSQL_CCFLAGS

PLSQL_CODE_TYPE

PLSQL_DEBUG

PLSQL_OPTIMIZE_LEVEL

PLSQL_V2_COMPATIBILITY

PLSQL_WARNINGS

QUERY_REWRITE_ENABLED

QUERY_REWRITE_INTEGRITY

RECYCLEBIN

REMOTE_DEPENDENCIES_MODE

```
RESULT_CACHE_MODE
RESULT_CACHE_REMOTE_EXPIRATION
RESUMABLE_TIMEOUT
SEC_PROTOCOL_ERROR_FURTHER_ACTION
SEC_PROTOCOL_ERROR_TRACE_ACTION
SESSION_CACHED_CURSORS
SKIP_UNUSABLE_INDEXES
SMTP_OUT_SERVER
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
SQL_TRACE
SQLTUNE_CATEGORY
STAR_TRANSFORMATION_ENABLED
STATISTICS_LEVEL
TIMED_OS_STATISTICS
TIMED_STATISTICS
TRACEFILE_IDENTIFIER
WORKAREA_SIZE_POLICY
XML_DB_EVENTS
```

The ALTER SYSTEM statement without the DEFERRED keyword modifies the global value of the parameter for all sessions in the instance, for the duration of the instance (until the database is shut down). The value of the following initialization parameters can be changed with ALTER SYSTEM:

```
AQ_TM_PROCESSES
ARCHIVE_LAG_TARGET
ASM_DISKGROUPS
ASM_DISKSTRING
ASM_POWER_LIMIT
ASM_PREFERRED_READ_FAILURE_GROUPS
BACKGROUND_DUMP_DEST
CIRCUITS
COMMIT_LOGGING
COMMIT_WAIT
COMMIT_WRITE
CONTROL_FILE_RECORD_KEEP_TIME
CONTROL_MANAGEMENT_PACK_ACCESS
CORE_DUMP_DEST
CPU_COUNT
CREATE_STORED_OUTLINES
CURSOR_SHARING
DB_nK_CACHE_SIZE
DB_BLOCK_CHECKING
DB_BLOCK_CHECKSUM
DB_CACHE_ADVICE
DB_CACHE_SIZE
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_FILE_MULTIBLOCK_READ_COUNT
DB_FLASHBACK_RETENTION_TARGET
DB_KEEP_CACHE_SIZE
DB_LOST_WRITE_PROTECT
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_RECYCLE_CACHE_SIZE
DB_SECUREFILE
```

```
DDL_LOCK_TIMEOUT
DG_BROKER_CONFIG_FILEn
DG_BROKER_START
DIAGNOSTIC_DEST
DISPATCHERS
FAL_CLIENT
FAL_SERVER
FAST_START_IO_TARGET
FAST_START_MTTR_TARGET
FAST_START_PARALLEL_ROLLBACK
FILE_MAPPING
FIXED_DATE
GLOBAL_NAMES
GLOBAL_TXN_PROCESSES
HS_AUTOREGISTER
JAVA_JIT_ENABLED
JAVA_POOL_SIZE
JOB_QUEUE_PROCESSES
LARGE_POOL_SIZE
LDAP_DIRECTORY_ACCESS
LICENSE_MAX_SESSIONS
LICENSE_MAX_USERS
LICENSE_SESSIONS_WARNING
LOCAL_LISTENER
LOG_ARCHIVE_CONFIG
LOG_ARCHIVE_DEST
LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n
LOG_ARCHIVE_DUPLEX_DEST
LOG_ARCHIVE_LOCAL_FIRST
LOG_ARCHIVE_MAX_PROCESSES
LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_ARCHIVE_TRACE
LOG_CHECKPOINT_INTERVAL
LOG_CHECKPOINT_TIMEOUT
LOG_CHECKPOINTS_TO_ALERT
MAX_DISPATCHERS
MAX_DUMP_FILE_SIZE
MAX_SHARED_SERVERS
MEMORY_TARGET
NLS_LENGTH_SEMANTICS
NLS_NCHAR_CONV_EXCP
OPEN_CURSORS
OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES
OPTIMIZER_DYNAMIC_SAMPLING
OPTIMIZER_FEATURES_ENABLE
OPTIMIZER_INDEX_CACHING
OPTIMIZER_INDEX_COST_ADJ
OPTIMIZER_MODE
OPTIMIZER_SECURE_VIEW_MERGING
OPTIMIZER_USE_INVISIBLE_INDEXES
OPTIMIZER_USE_PENDING_STATISTICS
OPTIMIZER_USE_SQL_PLAN_BASELINES
PARALLEL_ADAPTIVE_MULTI_USER
PARALLEL_INSTANCE_GROUP
```

```
PARALLEL_IO_CAP_ENABLED
PARALLEL_MAX_SERVERS
PARALLEL_MIN_SERVERS
PARALLEL_THREADS_PER_CPU
PGA_AGGREGATE_TARGET
PLSCOPE_SETTINGS
PLSQL_CCFLAGS
PLSQL_CODE_TYPE
PLSQL_DEBUG
PLSQL_NATIVE_LIBRARY_DIR
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
PLSQL_OPTIMIZE_LEVEL
PLSQL_V2_COMPATIBILITY
PLSQL_WARNINGS
QUERY_REWRITE_ENABLED
QUERY_REWRITE_INTEGRITY
REDO_TRANSPORT_USER
REMOTE_DEPENDENCIES_MODE
REMOTE_LISTENER
RESOURCE_LIMIT
RESOURCE_MANAGER_CPU_ALLOCATION
RESOURCE_MANAGER_PLAN
RESULT_CACHE_MAX_RESULT
RESULT_CACHE_MAX_SIZE
RESULT_CACHE_MODE
RESULT_CACHE_REMOTE_EXPIRATION
RESUMABLE_TIMEOUT
SEC_CASE_SENSITIVE_LOGON
SEC_PROTOCOL_ERROR_FURTHER_ACTION
SEC_PROTOCOL_ERROR_TRACE_ACTION
SERVICE_NAMES
SGA_TARGET
SHARED_POOL_SIZE
SHARED_SERVER_SESSIONS
SHARED_SERVERS
SKIP_UNUSABLE_INDEXES
SMTP_OUT_SERVER
SPFILE
SQL_TRACE
SQLTUNE_CATEGORY
STANDBY_ARCHIVE_DEST
STANDBY_FILE_MANAGEMENT
STAR_TRANSFORMATION_ENABLED
STATISTICS_LEVEL
STREAMS_POOL_SIZE
THREAD
TIMED_OS_STATISTICS
TIMED_STATISTICS
TRACE_ENABLED
UNDO_RETENTION
UNDO_TABLESPACE
USER_DUMP_DEST
WORKAREA_SIZE_POLICY
XML_DB_EVENTS
```

The ALTER SYSTEM ... DEFERRED statement does not modify the global value of the parameter for existing sessions, but the value will be modified for future sessions that connect to the database. The value of the following initialization parameters can be changed with ALTER SYSTEM ... DEFERRED:

```
AUDIT_FILE_DEST
BACKUP_TAPE_IO_SLAVES
OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE
OLAP_PAGE_POOL_SIZE
RECYCLEBIN
SORT_AREA_RETAINED_SIZE
SORT_AREA_SIZE
```

Displaying Current Parameter Values

To see the current settings for initialization parameters, use the following SQL*Plus command:

```
SQL> SHOW PARAMETERS
```

This command displays all parameters in alphabetical order, along with their current values.

Enter the following text string to display all parameters having BLOCK in their names:

```
SQL> SHOW PARAMETERS BLOCK
```

You can use the SPOOL command to write the output to a file.

Parameters You Should Not Specify in the Parameter File

You should not specify the following two types of parameters in your parameter files:

- Parameters that you never alter except when instructed to do so by Oracle to resolve a problem
- Derived parameters, which normally do not need altering because their values are calculated automatically by the Oracle database server

When Parameters Are Set Incorrectly

Some parameters have a minimum setting below which an Oracle instance will not start. For other parameters, setting the value too low or too high may cause Oracle to perform badly, but it will still run. Also, Oracle may convert some values outside the acceptable range to usable levels.

If a parameter value is too low or too high, or you have reached the maximum for some resource, then Oracle returns an error. Frequently, you can wait a short while and retry the operation when the system is not as busy. If a message occurs repeatedly, then you should shut down the instance, adjust the relevant parameter, and restart the instance.

Reading the Parameter Descriptions

The parameter descriptions in this chapter adhere to the following format.

PARAMETER_NAME

Property	Description
Parameter type	Specifies the type of the parameter: ▪ A Boolean parameter accepts either `true` or `false` as its value. ▪ A string parameter accepts any sequence of characters as its value, subject to the syntax for the parameter. ▪ An integer parameter accepts a 4-byte value that can range from 0 to 2^{32} - 1. ▪ A parameter file parameter accepts an initialization parameter file specification as its value. ▪ A **big integer** parameter accepts an 8-byte value that can range from 0 to 2^{64} - 1. You specify a value for a big integer as an integer together with an optional modifier such as K, M, or G, which respectively denotes kilobytes, megabytes, or gigabytes. For example, 1000, 100 KB, 50 MB and 2 GB are valid specifications for big integers.
Syntax	For string and big integer parameters, specifies the valid syntax for specifying the parameter.
Default value	Specifies the value this parameter assumes if not explicitly specified.
Modifiable	Specifies whether the parameter can be changed for the current session (by an `ALTER SESSION` statement) or for all sessions in the current instance (by an `ALTER SYSTEM` statement): ▪ `ALTER SESSION` overrides the instance-wide setting of the parameter for the current session only. You can restore the instance-wide setting for that session only by issuing another `ALTER SESSION` statement. ▪ `ALTER SYSTEM` can be used to change the value in the server parameter file (SPFILE) of any initialization parameter. Such a change takes effect only in subsequent instances. The parameter descriptions indicate only those parameters that can be modified for the current instance.
Range of values	Specifies the valid range of values that this parameter can assume, shown as a minimum and maximum value. Not applicable to all parameters.
Real Application Clusters	Specifies how the values for this parameter must be specified for multiple instances in a cluster database environment. Not applicable to all parameters.

For each parameter, paragraphs following these details further describe the parameter and the effects of different settings.

Initialization Parameter Descriptions

The remainder of this chapter describes the initialization parameters in alphabetical order. Initialization parameter values apply to the entire database, not to an individual user, unless otherwise specified.

Note: Parameters that have become obsolete are not documented.

See Also:

- *Oracle Database Upgrade Guide* for information about obsolete parameters
- Your system release bulletins or other operating system-specific Oracle documentation

ACTIVE_INSTANCE_COUNT

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	No
Range of values	1 or >= the number of instances in the cluster. (Values other than 1 have no effect on the active or standby status of any instances.)
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

Note: This parameter functions only in a cluster with only two instances.

`ACTIVE_INSTANCE_COUNT` enables you to designate one instance in a two-instance cluster as the primary instance and the other instance as the secondary instance. This parameter has no functionality in a cluster with more than two instances.

When you set this parameter to 1, the first instance you start up becomes the primary instance and accepts client connections. The second instance starts up as a secondary instance and can accept client connections only if the first instance fails. In such an event, the secondary instance becomes the primary instance.

When the failed instance can once again be started up, it starts up as the secondary instance, and will not accept client connections unless the current primary instance fails.

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system for more information on setting this parameter

AQ_TM_PROCESSES

Property	Description
Parameter type	Integer
Default value	None
Modifiable	`ALTER SYSTEM`

Property	Description
Range of values	`0` to `10`
Basic	No

`AQ_TM_PROCESSES` controls time monitoring on queue messages and controls processing of messages with delay and expiration properties specified.You do not need to specify a value for this parameter because Oracle Database automatically determines the number of processes and autotunes them, as necessary. Therefore, Oracle highly recommends that you leave the `AQ_TM_PROCESSES` parameter unspecified and let the system autotune.

> **Note:** If you want to disable the Queue Monitor Coordinator, then you must set `AQ_TM_PROCESSES` to 0 in your parameter file. Oracle strongly recommends that you do NOT set `AQ_TM_PROCESSES` to 0. If you are using Oracle Streams, then setting this parameter to zero (which Oracle Database respects no matter what) can cause serious problems.

> **See Also:** *Oracle Streams Advanced Queuing User's Guide* for more information about this parameter and Advanced Queuing

ARCHIVE_LAG_TARGET

Property	Description
Parameter type	Integer
Default value	`0` (disabled)
Modifiable	`ALTER SYSTEM`
Range of values	0 or any integer in [60, 7200]
Basic	No
Real Application Clusters	Multiple instances should use the same value.

`ARCHIVE_LAG_TARGET` limits the amount of data that can be lost and effectively increases the availability of the standby database by forcing a log switch after the specified amount of time elapses.

A 0 value disables the time-based thread advance feature; otherwise, the value represents the number of seconds. Values larger than 7200 seconds are not of much use in maintaining a reasonable lag in the standby database. The typical, or recommended value is 1800 (30 minutes). Extremely low values can result in frequent log switches, which could degrade performance; such values can also make the archiver process too busy to archive the continuously generated logs.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about criteria for setting this parameter

ASM_DISKGROUPS

Property	Description
Parameter type	String
Syntax	`ASM_DISKGROUPS =` *diskgroup* `[,` *diskgroup* `] ...`
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Range of values	Comma-separated list of strings, of up to 30 characters
Real Application Clusters	Multiple instances can have different values.

Note: This parameter may only be specified in an Automatic Storage Management instance.

`ASM_DISKGROUPS` specifies a list of names of disk groups to be mounted by an Automatic Storage Management instance at instance startup or when an `ALTER DISKGROUP ALL MOUNT` statement is issued.

Automatic Storage Management (ASM) automatically adds a disk group to this parameter when the disk group is successfully created or mounted, and automatically removes a disk group from this parameter when the disk group is dropped or dismounted.

Issuing the `ALTER DISKGROUP...ALL MOUNT` or `ALTER DISKGROUP...ALL DISMOUNT` command does not affect the value of this parameter.

ASM_DISKSTRING

Property	Description
Parameter type	String
Syntax	`ASM_DISKSTRING =` *discovery_string* `[,` *discovery_string* `] ...`
Default value	The null string; Automatic Storage Management discovery finds all disks in an operating system-specific location to which the Automatic Storage Management instance has read/write access.
Modifiable	`ALTER SYSTEM`
Real Application Clusters	Multiple instances can have different values. Different nodes might see the same disks under different names; however, each instance must be able to use its `ASM_DISKSTRING` to discover the same physical media as the other nodes in the cluster.

Note: This parameter may only be specified in an Automatic Storage Management instance.

`ASM_DISKSTRING` specifies an operating system-dependent value used by Automatic Storage Management to limit the set of disks considered for discovery. When a new disk is added to a disk group, each Automatic Storage Management instance that has the disk group mounted must be able to discover the new disk using the value of ASM_DISKSTRING.

In most cases, the default value will be sufficient. Using a more restrictive value may reduce the time required for Automatic Storage Management to perform discovery, and thus improve disk group mount time or the time for adding a disk to a disk group. A "?" at the beginning of the string gets expanded to the Oracle home directory. Depending on the operating system, wildcard characters can be used.

It may be necessary to dynamically change ASM_DISKSTRING before adding a disk so that the new disk will be discovered.

An attempt to dynamically modify ASM_DISKSTRING will be rejected and the old value retained if the new value cannot be used to discover a disk that is in a disk group that is already mounted.

ASM_POWER_LIMIT

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to 11
Real Application Clusters	Multiple instances can have different values.

Note: This parameter may only be specified in an Automatic Storage Management instance.

ASM_POWER_LIMIT specifies the maximum power on an Automatic Storage Management instance for disk rebalancing. The higher the limit, the faster rebalancing will complete. Lower values will take longer, but consume fewer processing and I/O resources.

If the POWER clause of a rebalance operation is not specified, then the default power will be the value of ASM_POWER_LIMIT.

ASM_PREFERRED_READ_FAILURE_GROUPS

Property	Description
Parameter type	String
Syntax	ASM_PREFERRED_READ_FAILURE_GROUPS = <diskgroup_name>.<failure_group_name>, ...
Default value	NULL
Modifiable	ALTER SYSTEM
Basic	No
Real Application Clusters	The value is different on different nodes

ASM_PREFERRED_READ_FAILURE_GROUPS specifies the failure groups that contain preferred read disks. Preferred disks are instance specific. This parameter is only valid in ASM instances.

AUDIT_FILE_DEST

Property	Description
Parameter type	String
Syntax	AUDIT_FILE_DEST = '*directory*'
Default value	The first default value is: *ORACLE_BASE*/admin/*ORACLE_SID/adump* The second default value, which is used if the first default value does not exist or is unusable, is: *ORACLE_HOME*/rdbms/audit Both of these default values are for Unix systems. Other platforms may have different defaults.
Modifiable	ALTER SYSTEM ... DEFERRED
Basic	No

AUDIT_FILE_DEST specifies the operating system directory into which the audit trail is written when the AUDIT_TRAIL initialization parameter is set to **os**, **xml**, or **xml,extended**. The audit records will be written in XML format if the AUDIT_TRAIL initialization parameter is set to XML. It is also the location to which mandatory auditing information is written and, if so specified by the AUDIT_SYS_OPERATIONS initialization parameter, audit records for user SYS.

AUDIT_SYS_OPERATIONS

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true | false
Basic	No

AUDIT_SYS_OPERATIONS enables or disables the auditing of operations issued by user SYS, and users connecting with SYSDBA or SYSOPER privileges. The audit records are written to the operating system's audit trail. The audit records will be written in XML format if the AUDIT_TRAIL initialization parameter is set to XML.

On UNIX platforms, if the AUDIT_SYSLOG_LEVEL parameter has also been set, then it overrides the AUDIT_TRAIL parameter and SYS audit records are written to the system audit log using the SYSLOG utility.

AUDIT_SYSLOG_LEVEL

Property	Description
Parameter type	String

Syntax	`AUDIT_SYSLOG_LEVEL = 'facility_clause.priority_clause'`
	facility_clause::=
	`{ USER | LOCAL[0 | 1 | 2 | 3 | 4 | 5 | 6 | 7] | SYSLOG | DAEMON | KERN | MAIL | AUTH | LPR | NEWS | UUCP | CRON }`
	priority_clause::=
	`{ NOTICE | INFO | DEBUG | WARNING | ERR | CRIT | ALERT | EMERG }`

Default value	There is no default value.
Modifiable	No
Basic	No

Examples	`AUDIT_SYSLOG_LEVEL = 'KERN.EMERG';`
	`AUDIT_SYSLOG_LEVEL = 'LOCAL1.WARNING';`

`AUDIT_SYSLOG_LEVEL` allows `SYS` and standard `OS` audit records to be written to the system audit log using the `SYSLOG` utility.

If you use this parameter, it is best to assign a file corresponding to every combination of facility and priority (especially `KERN.EMERG`) in `syslog.conf` . Sometimes these are assigned to print to the console in the default `syslog.conf` file. This can become annoying and will be useless as audit logs. Also, if you use this parameter, it is best to set the maximum length of syslog messages in the system to 512 bytes.

If `AUDIT_SYSLOG_LEVEL` is set and `SYS` auditing is enabled (`AUDIT_SYS_OPERATIONS` = `TRUE`), then `SYS` audit records are written to the system audit log.If `AUDIT_SYSLOG_LEVEL` is set and standard audit records are being sent to the operating system (`AUDIT_TRAIL` = `OS`), then standard audit records are written to the system audit log.

> **See Also:** *Oracle Database Security Guide* for more information about configuring the syslog audit trail by using the `AUDIT_SYSLOG_LEVEL` parameter

AUDIT_TRAIL

Property	Description
Parameter type	String
Syntax	`AUDIT_TRAIL = { none | os | db | db_extended | xml | xml, extended }`
Default value	`db`
Modifiable	No
Basic	No

`AUDIT_TRAIL` enables or disables database auditing.

Values:

- `none`

  Disables database auditing.

- `os`

  Enables database auditing and directs all audit records to the operating system's audit trail.

- `db`

  Enables database auditing and directs all audit records to the database audit trail (the `SYS.AUD$` table).

- `db_extended`

  Enables database auditing and directs all audit records to the database audit trail (the `SYS.AUD$` table). In addition, populates the `SQLBIND` and `SQLTEXT` CLOB columns of the `SYS.AUD$` table.

- `xml`

  Enables database auditing and writes all audit records to XML format OS files.

- `xml, extended`

  Enables database auditing and prints all columns of the audit trail, including SqlText and SqlBind values.

You can use the SQL `AUDIT` statement to set auditing options regardless of the setting of this parameter.

BACKGROUND_CORE_DUMP

Property	Description
Parameter type	String
Syntax	`BACKGROUND_CORE_DUMP = { partial | full }`
Default value	`partial`
Modifiable	No
Basic	No

`BACKGROUND_CORE_DUMP` specifies whether Oracle includes the SGA in the core file for Oracle background processes.

Values:

- `partial`

  Oracle does not include the SGA in the core dump.

- `full`

  Oracle includes the SGA in the core dump.

See Also: "SHADOW_CORE_DUMP" on page 1-147

BACKGROUND_DUMP_DEST

Property	Description
Parameter type	String
Syntax	`BACKGROUND_DUMP_DEST = {` *`pathname`* `|` *`directory`* `}`
Default value	Operating system-dependent
Modifiable	`ALTER SYSTEM`
Range of values	Any valid local path, directory, or disk
Basic	No

`BACKGROUND_DUMP_DEST` specifies the pathname (directory or disc) where debugging trace files for the background processes (LGWR, DBW*n*, and so on) are written during Oracle operations.

An **alert file** in the directory specified by `BACKGROUND_DUMP_DEST` logs significant database events and messages. Anything that affects the database instance or global database is recorded here. The alert file is a normal text file. Its filename is operating system-dependent. For platforms that support multiple instances, it takes the form `alert_`*`sid`*`.log`, where *sid* is the system identifier. This file grows slowly, but without limit, so you might want to delete it periodically. You can delete the file even when the database is running.

Note: This parameter is ignored by the new diagnosability infrastructure introduced in Oracle Database 11g Release 1, which places trace and core files in a location controlled by the `DIAGNOSTIC_DEST` initialization parameter.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter and on alert files
- Your operating system-specific Oracle documentation for the default value of this parameter
- "USER_DUMP_DEST" on page 1-165 for information on setting a destination for server process trace files

BACKUP_TAPE_IO_SLAVES

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SYSTEM ... DEFERRED`
Range of values	`true	false`
Basic	No

`BACKUP_TAPE_IO_SLAVES` specifies whether I/O server processes (also called **slaves**) are used by Recovery Manager to back up, copy, or restore data to tape. When the value is set to `true`, Oracle uses an I/O server process to write to or read from a tape device. When the value is set to `false` (the default), Oracle does not use I/O server process for backups. Instead, the shadow process engaged in the backup accesses the tape device.

Note: You cannot perform duplexed backups unless you enable this parameter. Otherwise, Oracle returns an error. When this parameter is enabled, Recovery Manager will configure as many server processes as needed for the number of backup copies requested.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information on duplexed backups
- "DBWR_IO_SLAVES" on page 1-54

BITMAP_MERGE_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	`1048576` (1 MB)
Modifiable	No
Range of values	Operating system-dependent
Basic	No

Note: Oracle does not recommend using the `BITMAP_MERGE_AREA_SIZE` parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting `PGA_AGGREGATE_TARGET` instead. `BITMAP_MERGE_AREA_SIZE` is retained for backward compatibility.

`BITMAP_MERGE_AREA_SIZE` is relevant only for systems containing bitmap indexes. It specifies the amount of memory Oracle uses to merge bitmaps retrieved from a range scan of the index. The default value is 1 MB. A larger value usually improves performance, because the bitmap segments must be sorted before being merged into a single bitmap.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

BLANK_TRIMMING

Property	Description
Parameter type	Boolean

Property	Description
Default value	`false`
Modifiable	No
Range of values	`true` | `false`
Basic	No

`BLANK_TRIMMING` specifies the data assignment semantics of character datatypes.

Values:

- `TRUE`

  Allows the data assignment of a source character string or variable to a destination character column or variable even though the source length is longer than the destination length. In this case, however, the additional length over the destination length is all blanks. This value complies with the semantics of SQL92 Transitional Level and above.

- `FALSE`

  Disallows the data assignment if the source length is longer than the destination length and reverts to SQL92 Entry Level semantics.

CIRCUITS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Basic	No

`CIRCUITS` specifies the total number of virtual circuits that are available for inbound and outbound network sessions. It is one of several parameters that contribute to the total SGA requirements of an instance.

You should not specify a value for this parameter unless you want to limit the number of virtual circuits.

> **See Also:** *Oracle Database Concepts* for more information on memory structures and processes

CLIENT_RESULT_CACHE_LAG

Property	Description
Parameter type	Big integer
Syntax	`CLIENT_RESULT_CACHE_LAG =` *`integer`* `[K	M	G]`
Default value	`5000`
Modifiable	No
Range of values	`0` to operating system-dependent

Property	Description
Basic	No

`CLIENT_RESULT_CACHE_LAG` specifies the maximum time (in milliseconds) since the last round trip to the server, before which the OCI client query execute makes a round trip to get any database changes related to the queries cached on the client.

> **See Also:** *Oracle Call Interface Programmer's Guide* for more information about the client query cache feature

CLIENT_RESULT_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	`CLIENT_RESULT_CACHE_SIZE = *integer* [K | M | G]`
Default value	`0`
Modifiable	No
Range of values	`0` to operating system-dependent
Basic	No

`CLIENT_RESULT_CACHE_SIZE` specifies the maximum size of the client per-process result set cache (in bytes). All OCI client processes inherit this maximum size. Setting a nonzero value enables the client query cache feature. This can be overridden by the client configuration parameter `OCI_RESULT_CACHE_MAX_SIZE`.

> **See Also:** *Oracle Call Interface Programmer's Guide* for more information about the client query cache feature

CLUSTER_DATABASE

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true | false`
Basic	Yes
Real Application Clusters	For all instances, the value must be set to `TRUE`.

`CLUSTER_DATABASE` is a Real Application Clusters parameter that specifies whether or not Real Application Clusters is enabled.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide*

CLUSTER_DATABASE_INSTANCES

Property	Description
Parameter type	Integer
Default value	`1`
Modifiable	No
Range of values	Any nonzero value
Basic	No
Real Application Clusters	Multiple instances must have the same value.

`CLUSTER_DATABASE_INSTANCES` is a Real Application Clusters parameter that specifies the number of instances that are configured as part of your cluster database. You must set this parameter for every instance. Normally you should set this parameter to the number of instances in your Real Application Clusters environment. A proper setting for this parameter can improve memory use.

Oracle uses the value of this parameter to compute the default value of the `LARGE_POOL_SIZE` parameter when the `PARALLEL_AUTOMATIC_TUNING` parameter is set to `true`. Note that the `PARALLEL_AUTOMATIC_TUNING` parameter has been deprecated.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution
- *Oracle Real Application Clusters Administration and Deployment Guide* for information on Real Application Clusters

CLUSTER_INTERCONNECTS

Property	Description
Parameter type	String
Syntax	`CLUSTER_INTERCONNECTS =` *`ifn`* `[:` *`ifn`* `] ...`
Default value	There is no default value.
Modifiable	No
Range of values	One or more IP addresses, separated by colons
Basic	No

`CLUSTER_INTERCONNECTS` provides Oracle with information about additional cluster interconnects available for use in Real Application Clusters environments.

The `CLUSTER_INTERCONNECTS` parameter can be used to override the default interconnect with a preferred cluster traffic network. This parameter is useful in Data Warehouse systems that have reduced availability requirements and high interconnect bandwidth demands. You can also use `CLUSTER_INTERCONNECTS` to override the default interconnect chosen by Oracle.

For example, if you are running two instances of Oracle for two databases on the same machine, then you can load balance the interconnect traffic to different physical interconnects. This does not reduce Oracle availability.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide*

COMMIT_LOGGING

Property	Description
Parameter type	String
Syntax	`COMMIT_LOGGING = { IMMEDIATE | BATCH }`
Default value	There is no default value.
Modifiable	Yes (at both session-level and system-level)
Basic	No
Real Application Clusters	Each instance may have its own setting

`COMMIT_LOGGING` is an advanced parameter used to control how redo is batched by Log Writer.

If `COMMIT_LOGGING` is altered after setting `COMMIT_WAIT` to `FORCE_WAIT`, then the `FORCE_WAIT` option is no longer valid.

COMMIT_POINT_STRENGTH

Property	Description
Parameter type	Integer
Default value	`1`
Modifiable	No
Range of values	`0` to `255`
Basic	No

`COMMIT_POINT_STRENGTH` is relevant only in distributed database systems. It specifies a value that determines the **commit point site** in a distributed transaction. The node in the transaction with the highest value for `COMMIT_POINT_STRENGTH` will be the commit point site.

The commit point site of a database should reflect the amount of critical shared data in the database. For example, a database on a mainframe computer typically shares more data among users than one on a personal computer. Therefore, `COMMIT_POINT_STRENGTH` should be set to a higher value for the mainframe computer.

The commit point site stores information about the status of transactions. Other computers in a distributed transaction require this information during Oracle's two-phase commit, so it is desirable to have machines that are always available as commit point sites. Therefore, set `COMMIT_POINT_STRENGTH` to a higher value on your more available machines.

See Also:

- *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for information on two-phase commit
- Your operating system-specific Oracle documentation for the default value of this parameter

COMMIT_WAIT

Property	Description
Parameter type	String
Syntax	`COMMIT_WAIT = { NOWAIT | WAIT | FORCE_WAIT }`
Default value	There is no default value.
Modifiable	Yes (at both session-level and system-level)
Basic	No
Real Application Clusters	Each instance may have its own setting

`COMMIT_WAIT` is an advanced parameter used to control when the redo for a commit is flushed to the redo logs.

If the parameter is set to `FORCE_WAIT`, the default behavior (immediate flushing of the redo log buffer with wait) is used. If this is a system setting, the session level and transaction level (`COMMIT_WRITE`) options will be ignored. If this is a session level setting, the transaction level options will be ignored. If `COMMIT_WAIT` is altered after it has been set, then the `FORCE_WAIT` option is no longer valid.

COMMIT_WRITE

Property	Description
Parameter type	String
Syntax	`COMMIT_WRITE = '{IMMEDIATE | BATCH},{WAIT |NOWAIT}'`
Default value	If this parameter is not explicitly specified, then database commit behavior defaults to writing commit records to disk before control is returned to the client. If only `IMMEDIATE` or `BATCH` is specified, but not `WAIT` or `NOWAIT`, then `WAIT` mode is assumed. If only `WAIT` or `NOWAIT` is specified, but not `IMMEDIATE` or `BATCH`, then `IMMEDIATE` mode is assumed.
Modifiable	Yes (at both session-level and system-level). Values supplied for `COMMIT_WRITE` in an `ALTER SYSTEM` or `ALTER SESSION` statement must be separated by a comma.
Range of values	Single-quoted, comma-separated list of either `IMMEDIATE` or `BATCH`, and either `WAIT` or `NOWAIT`.
Basic	No
Real Application Clusters	Each instance may have its own setting

COMMIT_WRITE is an advanced parameter used to control how redo for transaction commits is written to the redo logs. The IMMEDIATE and BATCH options control how redo is batched by Log Writer. The WAIT and NOWAIT options control when the redo for a commit is flushed to the redo logs.

> **Note:** The COMMIT_WRITE parameter is deprecated. It is retained for backward compatibility only. It is replaced by the COMMIT_LOGGING and COMMIT_WAIT parameters.

COMPATIBLE

Property	Description
Parameter type	String
Syntax	COMPATIBLE = *release_number*
Default value	11.0.0
Modifiable	No
Range of values	10.0.0 to default release
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

COMPATIBLE allows you to use a new release of Oracle, while at the same time guaranteeing backward compatibility with an earlier release. This is helpful if it becomes necessary to revert to the earlier release.

This parameter specifies the release with which Oracle must maintain compatibility. It allows you to take advantage of the maintenance improvements of a new release immediately in your production systems without testing the new functionality in your environment. Some features of the release may be restricted.

When using a standby database, this parameter must have the same value on both the primary and standby databases.

> **See Also:** *Oracle Database Upgrade Guide* for more information on setting this parameter

CONTROL_FILE_RECORD_KEEP_TIME

Property	Description
Parameter type	Integer
Default value	7 (days)
Modifiable	ALTER SYSTEM
Range of values	0 to 365 (days)
Basic	No

CONTROL_FILE_RECORD_KEEP_TIME specifies the minimum number of days before a reusable record in the control file can be reused. In the event a new record needs to be added to a reusable section and the oldest record has not aged enough, the record section expands. If this parameter is set to 0, then reusable sections never expand, and records are reused as needed.

> **Note:** This parameter applies only to records in the control file that are circularly reusable (such as archive log records and various backup records). It does not apply to records such as datafile, tablespace, and redo thread records, which are never reused unless the corresponding object is dropped from the tablespace.

See Also:

- "CONTROL_FILES" on page 1-34
- *Oracle Database Backup and Recovery Basics*

CONTROL_FILES

Property	Description
Parameter type	String
Syntax	`CONTROL_FILES = *filename* [, *filename* ] ...` **Note:** The control file name can be an OMF (Oracle Managed Files) name. This occurs when the control file is re-created using the `CREATE CONTROLFILE REUSE` statement.
Default value	Operating system-dependent
Modifiable	No
Range of values	1 to 8 filenames
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

Every database has a **control file**, which contains entries that describe the structure of the database (such as its name, the timestamp of its creation, and the names and locations of its datafiles and redo files). `CONTROL_FILES` specifies one or more names of control files, separated by commas.

Oracle recommends that you multiplex multiple control files on different devices or mirror the file at the operating system level.

See Also:

- *Oracle Database Performance Tuning Guide*
- *Oracle Database Administrator's Guide*

CONTROL_MANAGEMENT_PACK_ACCESS

Property	Description
Parameter type	String

Property	Description
Syntax	`CONTROL_MANAGEMENT_PACK_ACCESS = { NONE | DIAGNOSTIC | DIAGNOSTIC+TUNING }`
Default value	`DIAGNOSTIC+TUNING`
Modifiable	`ALTER SYSTEM`
Basic	No

`CONTROL_MANAGEMENT_PACK_ACCESS` specifies which of the Server Manageability Packs should be active. The following packs are available:

- The `DIAGNOSTIC` pack includes AWR, ADDM, and so on.
- The `TUNING` pack includes SQL Tuning Advisor, SQLAccess Advisor, and so on.

  A license for `DIAGNOSTIC` is required for enabling the `TUNING` pack.

Values:

- `NONE`

  Both packs are not available.

- `DIAGNOSTIC`

  Only the `DIAGNOSTIC` pack is available.

- `DIAGNOSTIC+TUNING`

  Both packs are available. This is the default.

See Also: *Oracle Database Licensing Information*

CORE_DUMP_DEST

Property	Description
Parameter type	String
Syntax	`CORE_DUMP_DEST =` *`directory`*
Default value	*ORACLE_HOME*/DBS
Modifiable	`ALTER SYSTEM`
Basic	No

`CORE_DUMP_DEST` is primarily a UNIX parameter and may not be supported on your platform. It specifies the directory where Oracle dumps core files.

Note: This parameter is ignored by the new diagnosability infrastructure introduced in Oracle Database 11g Release 1, which places trace and core files in a location controlled by the `DIAGNOSTIC_DEST` initialization parameter.

CPU_COUNT

Property	Description
Parameter type	Integer
Default value	Set automatically by Oracle
Modifiable	ALTER SYSTEM Oracle strongly recommends that you do not alter the value of this parameter. If it is necessary to do so, please refer to your operating system-specific documentation for further information.
Range of values	0 to unlimited
Basic	No

Caution: On most platforms, Oracle automatically sets the value of CPU_COUNT to the number of CPUs available to your Oracle instance. Do not change the value of CPU_COUNT.

CPU_COUNT specifies the number of CPUs available to Oracle. On single-CPU computers, the value of CPU_COUNT is 1.

CREATE_BITMAP_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	8388608 (8 MB)
Modifiable	No
Range of values	Operating system-dependent
Basic	No

Note: Oracle does not recommend using the CREATE_BITMAP_AREA_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. CREATE_BITMAP_AREA_SIZE is retained for backward compatibility.

CREATE_BITMAP_AREA_SIZE is relevant only for systems containing bitmap indexes. It specifies the amount of memory (in bytes) allocated for bitmap creation. The default value is 8 MB. A larger value may speed up index creation.

Cardinality is the number of unique values in a column in relation to the number of rows in the table. If cardinality is very small, you can set a small value for this parameter. For example, if cardinality is only 2, then the value can be on the order of kilobytes rather than megabytes. As a general rule, the higher the cardinality, the more memory is needed for optimal performance.

See Also: *Oracle Database Performance Tuning Guide* for more information on using bitmap indexes

CREATE_STORED_OUTLINES

Property	Description
Parameter type	String
Syntax	`CREATE_STORED_OUTLINES = { true | false | *category_name* } [NOOVERRIDE]`
Default value	There is no default value.
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`CREATE_STORED_OUTLINES` determines whether Oracle automatically creates and stores an outline for each query submitted during the session.

Values:

- `true`

  Enables automatic outline creation for subsequent queries in the same session. These outlines receive a unique system-generated name and are stored in the `DEFAULT` category. If a particular query already has an outline defined for it in the `DEFAULT` category, then that outline will remain and a new outline will not be created.

- `false`

  Disables automatic outline creation during the session. This is the default.

- *`category_name`*

  Enables the same behavior as `true` except that any outline created during the session is stored in the *`category_name`* category.

- `NOOVERRIDE`

  `NOOVERRIDE` specifies that this system setting will not override the setting for any session in which this parameter was explicitly set. If you do not specify `NOOVERRIDE`, then this setting takes effect in all sessions.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

CURSOR_SHARING

Property	Description
Parameter type	String
Syntax	`CURSOR_SHARING = { SIMILAR | EXACT | FORCE }`
Default value	`EXACT`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

CURSOR_SHARING determines what kind of SQL statements can share the same cursors.

Values:

- FORCE

  Forces statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect the meaning of the statement.

- SIMILAR

  Causes statements that may differ in some literals, but are otherwise identical, to share a cursor, unless the literals affect either the meaning of the statement or the degree to which the plan is optimized.

- EXACT

  Only allows statements with identical text to share the same cursor.

> **Note:** Forcing cursor sharing among similar (but not identical) statements can have unexpected results in some DSS applications, or applications that use stored outlines.

> **See Also:** *Oracle Database Performance Tuning Guide* for information on setting this parameter in these and other environments

CURSOR_SPACE_FOR_TIME

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true | false
Basic	No

CURSOR_SPACE_FOR_TIME lets you use more space for cursors in order to save time. It affects both the shared SQL area and the client's private SQL area.

Values:

- TRUE

  Shared SQL areas are kept pinned in the shared pool. As a result, shared SQL areas are not aged out of the pool as long as an open cursor references them. Because each active cursor's SQL area is present in memory, execution is faster. However, the shared SQL areas never leave memory while they are in use. Therefore, you should set this parameter to TRUE only when the shared pool is large enough to hold all open cursors simultaneously.

  In addition, a setting of TRUE retains the private SQL area allocated for each cursor between executions instead of discarding it after cursor execution, saving cursor allocation and initialization time

- FALSE

Shared SQL areas can be deallocated from the library cache to make room for new SQL statements.

> **Note:** If this parameter is set to TRUE, the SERIAL_REUSE parameter is disabled.

See Also:

- *Oracle Database Concepts* and *Oracle Database Performance Tuning Guide*
- "SERIAL_REUSE" on page 1-142

DB_*n*K_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	`DB_[2 | 4 | 8 | 16 | 32]K_CACHE_SIZE = *integer* [K | M | G]`
Default value	`0` (additional block size caches are not configured by default)
Modifiable	`ALTER SYSTEM`
Range of values	Minimum: `0` (values greater than zero are automatically modified to be either the user-specified size rounded up to the granule size or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

`DB_nK_CACHE_SIZE` (where *n* = 2, 4, 8, 16, 32) specifies the size of the cache for the `nK` buffers. You can set this parameter only when `DB_BLOCK_SIZE` has a value other than `nK`. For example, if `DB_BLOCK_SIZE=4096`, then it is illegal to specify the parameter `DB_4K_CACHE_SIZE` (because the size for the 4 KB block cache is already specified by `DB_CACHE_SIZE`).

Do not set this parameter to zero if there are any online tablespaces with an `nK` block size.

Operating system-specific block size restrictions apply. For example, you cannot set `DB_32K_CACHE_SIZE` if the operating system's maximum block size is less than 32 KB. Also, you cannot set `DB_2K_CACHE_SIZE` if the minimum block size is greater than 2 KB.

> **See Also:** Your operating system-specific Oracle documentation for more information on block size restrictions

DB_BLOCK_BUFFERS

Property	Description
Parameter type	Integer
Default value	`0`

Property	Description
Modifiable	No
Range of values	50 to an operating system-specific maximum
Basic	No
Real Application Clusters	Multiple instances can have different values, and you can change the values as needed.

> **Note:** DB_BLOCK_BUFFERS cannot be combined with the dynamic DB_CACHE_SIZE parameter; combining these parameters in the same parameter file will produce an error.

DB_BLOCK_BUFFERS specifies the number of database buffers in the buffer cache. It is one of several parameters that contribute to the total memory requirements of the SGA of an instance.

This parameter, together with DB_BLOCK_SIZE, determines the total size of the buffer cache. Effective use of the buffer cache can greatly reduce the I/O load on the database. DB_BLOCK_SIZE can be specified only when the database is first created, so you use DB_BLOCK_BUFFERS to control the size of the buffer cache.

DB_BLOCK_CHECKING

Property	Description
Parameter type	String
Syntax	DB_BLOCK_CHECKING = { FALSE | OFF | LOW | MEDIUM | TRUE | FULL }
Default value	FALSE
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

DB_BLOCK_CHECKING specifies whether or not Oracle performs block checking for database blocks.

Values:

- OFF or FALSE

  No block checking is performed for blocks in user tablespaces. However, semantic block checking for SYSTEM tablespace blocks is always turned on.

- LOW

  Basic block header checks are performed after block contents change in memory (for example, after UPDATE or INSERT statements, on-disk reads, or inter-instance block transfers in Oracle RAC).

- MEDIUM

  All LOW checks are performed, as well as semantic block checking for all non-index-organized table blocks.

- FULL or TRUE

All LOW and MEDIUM checks are performed, as well as semantic checks for index blocks (that is, blocks of subordinate objects that can actually be dropped and reconstructed when faced with corruption).

Oracle checks a block by going through the data in the block, making sure it is logically self-consistent. Block checking can often prevent memory and data corruption. Block checking typically causes 1% to 10% overhead, depending on workload and the parameter value. The more updates or inserts in a workload, the more expensive it is to turn on block checking. You should set DB_BLOCK_CHECKING to FULL if the performance overhead is acceptable.

For backward compatibility the use of FALSE (implying OFF) and TRUE (implying FULL) is preserved.

DB_BLOCK_CHECKSUM

Property	Description
Parameter type	String
Syntax	DB_BLOCK_CHECKSUM = { OFF | FALSE | TYPICAL | TRUE | FULL }
Default value	TYPICAL
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

DB_BLOCK_CHECKSUM determines whether DBW*n* and the direct loader will calculate a checksum (a number calculated from all the bytes stored in the block) and store it in the cache header of every data block when writing it to disk. Checksums are verified when a block is read - only if this parameter is TYPICAL or FULL and the last write of the block stored a checksum. In FULL mode, Oracle also verifies the checksum before a change application from update/delete statements and recomputes it after the change is applied. In addition, Oracle gives every log block a checksum before writing it to the current log.

Starting with Oracle Database 11*g*, most of the log block checksum is done by the generating foreground processes, while the LGWR performs the rest of the work, for better CPU and cache efficiency. Prior to Oracle Database 11*g*, the LGWR solely performed the log block checksum. When this parameter is set to FULL, the LGWR verifies the checksum of each log block generated by the foreground processes before writing it to disk.

If this parameter is set to OFF, DBW*n* calculates checksums only for the SYSTEM tablespace, but not for user tablespaces. In addition, no log checksum is performed when this parameter is set to OFF.

Checksums allow Oracle to detect corruption caused by underlying disks, storage systems, or I/O systems. If set to FULL, DB_BLOCK_CHECKSUM also catches in-memory corruptions and stops them from making it to the disk. Turning on this feature in TYPICAL mode causes only an additional 1% to 2% overhead. In the FULL mode it causes 4% to 5% overhead. Oracle recommends that you set DB_BLOCK_CHECKSUM to TYPICAL.

For backward compatibility the use of TRUE (implying TYPICAL) and FALSE (implying OFF) values is preserved.

DB_BLOCK_SIZE

Property	Description
Parameter type	Integer
Default value	`8192`
Modifiable	No
Range of values	`2048` to `32768`, but your operating system may have a narrower range
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

> **Caution:** Set this parameter at the time of database creation. Do not alter it afterward.

`DB_BLOCK_SIZE` specifies (in bytes) the size of Oracle database blocks. Typical values are `4096` and `8192`. The value of this parameter must be a multiple of the physical block size at the device level.

The value for `DB_BLOCK_SIZE` in effect at the time you create the database determines the size of the blocks. The value must remain set to its initial value.

For Real Application Clusters, this parameter affects the maximum value of the `FREELISTS` storage parameter for tables and indexes. Oracle uses one database block for each freelist group. Decision support system (DSS) and data warehouse database environments tend to benefit from larger block size values.

See Also:

- *Oracle Database Performance Tuning Guide* and *Oracle Database Administrator's Guide* for information on setting this parameter
- *Oracle Database SQL Language Reference* for information on freelist groups
- *Oracle Database Performance Tuning Guide* for more information on the DSS and data warehouse environments
- *Oracle Database Concepts* for general information on block sizes

DB_CACHE_ADVICE

Property	Description
Parameter type	String
Syntax	`DB_CACHE_ADVICE = { ON | READY | OFF }`
Default value	If `STATISTICS_LEVEL` is set to `TYPICAL` or `ALL`, then `ON` If `STATISTICS_LEVEL` is set to `BASIC`, then `OFF`
Modifiable	`ALTER SYSTEM`
Basic	No

DB_CACHE_ADVICE enables or disables statistics gathering used for predicting behavior with different cache sizes through the V$DB_CACHE_ADVICE performance view.

Values:

- OFF

  Advisory is turned off and the memory for the advisory is not allocated.

- READY

  Advisory is turned off but the memory for the advisory remains allocated. Allocating the memory before the advisory is actually turned on avoids the risk of an error when you switch the parameter to ON.

  If the parameter is switched to this state from ON, the contents of the view are preserved and the memory for the advisory is retained.

  If the parameter is switched to this state from OFF, you may get an error.

- ON

  Advisory is turned on. CPU and memory overheads are incurred. Attempting to set the parameter to this state when it is already in the OFF state may result in an error. Otherwise, the view (V$DB_CACHE_ADVICE) is reset and statistics are gathered to the newly refreshed view.

  If the parameter is in the READY state, you can set it to ON without any errors because the memory is already allocated. The view is reset and statistics are displayed in the newly refreshed view.

DB_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	DB_CACHE_SIZE = *integer* [K | M | G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set, then the default is either 48 MB or 4 MB * number of CPUs, whichever is greater
Modifiable	ALTER SYSTEM
Basic	No

DB_CACHE_SIZE specifies the size of the DEFAULT buffer pool for buffers with the primary block size (the block size defined by the DB_BLOCK_SIZE initialization parameter).

The value must be at least 4M * number of cpus * granule size (smaller values are automatically rounded up to this value). A user-specified value larger than this is rounded up to the nearest granule size. A value of zero is illegal because it is needed for the DEFAULT memory pool of the primary block size, which is the block size for the SYSTEM tablespace.

> **See Also:** *Oracle Database Performance Tuning Guide* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DB_CREATE_FILE_DEST

Property	Description
Parameter type	String
Syntax	`DB_CREATE_FILE_DEST` = *directory* | *disk group*
Default value	There is no default value.
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	Yes

`DB_CREATE_FILE_DEST` specifies the default location for Oracle-managed datafiles. This location is also used as the default location for Oracle-managed control files and online redo logs if none of the `DB_CREATE_ONLINE_LOG_DEST_`*n* initialization parameters are specified.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle-managed file.

> **See Also:** *Oracle Database Administrator's Guide* for more information on setting this parameter and on Oracle-managed files

DB_CREATE_ONLINE_LOG_DEST_*n*

Property	Description
Parameter type	String
Syntax	`DB_CREATE_ONLINE_LOG_DEST_[1	2	3	4	5] =` *directory* | *disk group*
Default value	There is no default value.
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	Yes

`DB_CREATE_ONLINE_LOG_DEST_`*n* (where *n* = 1, 2, 3, ... 5) specifies the default location for Oracle-managed control files and online redo logs. If more than one `DB_CREATE_ONLINE_LOG_DEST_`*n* parameter is specified, then the control file or online redo log is multiplexed across the locations of the other `DB_CREATE_ONLINE_LOG_DEST_`*n* parameters. One member of each online redo log is created in each location, and one control file is created in each location.

Specifying at least two parameters provides greater fault tolerance for the control files and online redo logs if one of the locations should fail.

If a file system directory is specified as the default location, then the directory must already exist; Oracle does not create it. The directory must have appropriate permissions that allow Oracle to create files in it. Oracle generates unique names for the files, and a file thus created is an Oracle-managed file.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter and on Oracle-managed files

DB_DOMAIN

Property	Description
Parameter type	String
Syntax	`DB_DOMAIN` = *`domain_name`*
Default value	There is no default value.
Modifiable	No
Range of values	Any legal string of name components, separated by periods and up to 128 characters long (including the periods). This value cannot be `NULL`.
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

In a distributed database system, `DB_DOMAIN` specifies the logical location of the database within the network structure. You should set this parameter if this database is or ever will be part of a distributed system. The value consists of the extension components of a global database name, consisting of valid identifiers (any alphanumeric ASCII characters), separated by periods. Oracle recommends that you specify `DB_DOMAIN` as a unique string for all databases in a domain.

This parameter allows one department to create a database without worrying that it might have the same name as a database created by another department. If one sales department's `DB_DOMAIN` is `JAPAN.ACME.COM`, then their `SALES` database (`SALES.JAPAN.ACME.COM`) is uniquely distinguished from another database with `DB_NAME` = `SALES` but with `DB_DOMAIN` = `US.ACME.COM`.

If you omit the domains from the name of a database link, Oracle expands the name by qualifying the database with the domain of your local database as it currently exists in the data dictionary, and then stores the link name in the data dictionary. The characters valid in a database domain name are: alphanumeric characters, underscore (_), and number sign (#).

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- The data dictionary view "GLOBAL_NAME" on page 5-76

DB_FILE_MULTIBLOCK_READ_COUNT

Property	Description
Parameter type	Integer
Default value	The default value corresponds to the maximum I/O size that can be efficiently performed and is platform-dependent
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	Operating system-dependent

Property	Description
Basic	No

`DB_FILE_MULTIBLOCK_READ_COUNT` is one of the parameters you can use to minimize I/O during table scans. It specifies the maximum number of blocks read in one I/O operation during a sequential scan. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multiblock read count, and whether parallel execution is being utilized for the operation.

As of Oracle Database 10*g* release 2, the default value of this parameter is a value that corresponds to the maximum I/O size that can be performed efficiently. This value is platform-dependent and is 1MB for most platforms.

Because the parameter is expressed in blocks, it will be set to a value that is equal to the maximum I/O size that can be performed efficiently divided by the standard block size. Note that if the number of sessions is extremely large the multiblock read count value is decreased to avoid the buffer cache getting flooded with too many table scan buffers.

Even though the default value may be a large value, the optimizer will not favor large plans if you do not set this parameter. It would do so only if you explicitly set this parameter to a large value.

Online transaction processing (OLTP) and batch environments typically have values in the range of 4 to 16 for this parameter. DSS and data warehouse environments tend to benefit most from maximizing the value of this parameter. The optimizer is more likely to choose a full table scan over an index if the value of this parameter is high.

The maximum value is the operating system's maximum I/O size expressed as Oracle blocks ((max I/O size)/`DB_BLOCK_SIZE`). If you set this parameter to a value greater than the maximum, Oracle uses the maximum.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter

DB_FILE_NAME_CONVERT

Property	Description
Parameter type	String

Property	Description

Syntax	`DB_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' , ...`
	Where:
	- *string1* is the pattern of the primary database filename
	- *string2* is the pattern of the standby database filename
	- *string3* is the pattern of the primary database filename
	- *string4* is the pattern of the standby database filename
	You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks.
	The following are example settings that are acceptable:
	`DB_FILE_NAME_CONVERT = '/dbs/t1/','/dbs/t1/s_','dbs/t2/ ','dbs/t2/s_'`

Default value	There is no default value.
Modifiable	`ALTER SESSION`
Basic	No

`DB_FILE_NAME_CONVERT` is useful for creating a duplicate database for recovery purposes. It converts the filename of a new datafile on the primary database to a filename on the standby database. If you add a datafile to the primary database, you must add a corresponding file to the standby database. When the standby database is updated, this parameter converts the datafile name on the primary database to the datafile name on the standby database. The file on the standby database must exist and be writable, or the recovery process will halt with an error.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is signalled during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalue parameters).

Set the value of this parameter to two strings. The first string is the pattern found in the datafile names on the primary database. The second string is the pattern found in the datafile names on the standby database.

You can also use `DB_FILE_NAME_CONVERT` to rename the datafiles in the clone control file when setting up a clone database during tablespace point-in-time recovery.

See Also:

- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Data Guard Concepts and Administration*

DB_FILES

Property	Description
Parameter type	Integer
Default value	`200`
Modifiable	No

Property	Description
Range of values	Minimum: the largest among the absolute file numbers of the datafiles in the database Maximum: operating system-dependent
Basic	No
Real Application Clusters	Multiple instances must have the same value.

`DB_FILES` specifies the maximum number of database files that can be opened for this database. The maximum valid value is the maximum number of files, subject to operating system constraint, that will ever be specified for the database, including files to be added by `ADD DATAFILE` statements.

If you increase the value of `DB_FILES`, then you must shut down and restart all instances accessing the database before the new value can take effect. If you have a primary and standby database, then they should have the same value for this parameter.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for information on setting this parameter in a Real Application Clusters environment
- Your operating system-specific Oracle documentation for the default value of this parameter

DB_FLASHBACK_RETENTION_TARGET

Property	Description
Parameter type	Integer
Default value	`1440` (minutes)
Modifiable	`ALTER SYSTEM ... SID='*'`
Range of values	`0` to 2^{31} - 1
Basic	No

`DB_FLASHBACK_RETENTION_TARGET` specifies the upper limit (in minutes) on how far back in time the database may be flashed back. How far back one can flashback a database depends on how much flashback data Oracle has kept in the flash recovery area.

DB_KEEP_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	`DB_KEEP_CACHE_SIZE =` *`integer`* `[K | M | G]`
Default value	`0` (`DB_KEEP_CACHE_SIZE` is not configured by default)
Modifiable	`ALTER SYSTEM`

Property	Description
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the user-specified size rounded up to the granule size or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

`DB_KEEP_CACHE_SIZE` specifies the size of the `KEEP` buffer pool. The size of the buffers in the `KEEP` buffer pool is the primary block size (the block size defined by the `DB_BLOCK_SIZE` initialization parameter).

See Also:

- "DB_RECYCLE_CACHE_SIZE" on page 1-51
- *Oracle Database Performance Tuning Guide* for information on setting these parameters and on using multiple buffer pools

DB_LOST_WRITE_PROTECT

Property	Description
Parameter type	String
Syntax	`DB_LOST_WRITE_PROTECT = { NONE | TYPICAL | FULL }`
Default value	`NONE`
Modifiable	`ALTER SYSTEM`
Basic	No
Real Application Clusters	In RAC instances, the parameter value is system-wide.

`DB_LOST_WRITE_PROTECT` enables or disables lost write detection. A data block lost write occurs when an I/O subsystem acknowledges the completion of the block write, while in fact the write did not occur in the persistent storage.

When the parameter is set to `TYPICAL` on the primary database, the instance logs buffer cache reads for read-write tablespaces in the redo log, which is necessary for detection of lost writes.

When the parameter is set to `FULL` on the primary database, the instance logs reads for read-only tablespaces as well as read-write tablespaces.

When the parameter is set to `TYPICAL` or `FULL` on the standby database or on the primary database during media recovery, the instance performs lost write detection.

When the parameter is set to `NONE` on either the primary database or the standby database, no lost write detection functionality is enabled.

DB_NAME

Property	Description
Parameter type	String

Property	Description
Syntax	`DB_NAME = database_name`
Default value	There is no default value.
Modifiable	No
Basic	Yes
Real Application Clusters	You must set this parameter for every instance. Multiple instances must have the same value, or the same value must be specified in the `STARTUP OPEN` SQL*Plus command or the `ALTER DATABASE MOUNT` SQL statement.

`DB_NAME` specifies a database identifier of up to 8 characters. This parameter must be specified and must correspond to the name specified in the `CREATE DATABASE` statement.

If you have multiple databases, the value of this parameter should match the Oracle instance identifier of each one to avoid confusion with other databases running on the system. The value of `DB_NAME` should be the same in both the standby and production initialization parameter files.

The database name specified in either the `STARTUP` command or the `ALTER DATABASE ... MOUNT` statement for each instance of the cluster database must correspond to the `DB_NAME` initialization parameter setting.

The following characters are valid in a database name: alphanumeric characters, underscore (_), number sign (#), and dollar sign ($). No other characters are valid. Oracle removes double quotation marks before processing the database name. Therefore you cannot use double quotation marks to embed other characters in the name. The database name is case insensitive.

> **See Also:** *Oracle Database Administrator's Guide* and *Oracle Real Application Clusters Administration and Deployment Guide* for more information on setting this parameter

DB_RECOVERY_FILE_DEST

Property	Description
Parameter type	String
Syntax	`DB_RECOVERY_FILE_DEST = directory | disk group`
Default value	There is no default value.
Modifiable	`ALTER SYSTEM ... SID='*'`
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

`DB_RECOVERY_FILE_DEST` specifies the default location for the flash recovery area. The flash recovery area contains multiplexed copies of current control files and online redo logs, as well as archived redo logs, flashback logs, and RMAN backups.

Specifying this parameter without also specifying the `DB_RECOVERY_FILE_DEST_SIZE` initialization parameter is not allowed.

See Also:

- *Oracle Database Backup and Recovery Basics* for information on setting up and configuring the flash recovery area
- "DB_RECOVERY_FILE_DEST_SIZE" on page 1-51

DB_RECOVERY_FILE_DEST_SIZE

Property	Description
Parameter type	Big integer
Syntax	`DB_RECOVERY_FILE_DEST_SIZE =` *integer* `[K	M	G]`
Default value	There is no default value.
Modifiable	`ALTER SYSTEM ... SID='*'`
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have the same value.

`DB_RECOVERY_FILE_DEST_SIZE` specifies (in bytes) the hard limit on the total space to be used by target database recovery files created in the flash recovery area.

Note that neither block 0 nor the OS block header of each Oracle file is included in this size. Allow an extra 10% for this data when computing the actual disk usage required for the flash recovery area.

See Also:

- *Oracle Database Backup and Recovery Basics* for information on setting up and configuring the flash recovery area
- "DB_RECOVERY_FILE_DEST" on page 1-50

DB_RECYCLE_CACHE_SIZE

Property	Description
Parameter type	Big integer
Syntax	`DB_RECYCLE_CACHE_SIZE =` *integer* `[K	M	G]`
Default value	`0` (`DB_RECYCLE_CACHE_SIZE` is not configured by default)
Modifiable	`ALTER SYSTEM`
Range of values	Minimum: `0` (values greater than zero are automatically modified to be either the user-specified size rounded up to the granule size or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

`DB_RECYCLE_CACHE_SIZE` specifies the size of the `RECYCLE` buffer pool. The size of the buffers in the `RECYCLE` pool is the primary block size (the block size defined by the `DB_BLOCK_SIZE` initialization parameter).

See Also:

- "DB_KEEP_CACHE_SIZE" on page 1-48
- *Oracle Database Performance Tuning Guide* for information on setting these parameters and on using multiple buffer pools

DB_SECUREFILE

Property	Description
Parameter type	String
Syntax	`DB_SECUREFILE = { NEVER | PERMITTED | ALWAYS | IGNORE }`
Default value	`PERMITTED`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`DB_SECUREFILE` specifies whether or not to treat LOB files as SecureFiles.

Values:

- `NEVER`

  Any LOBs that are specified as SecureFiles are created as BasicFile LOBs. All SecureFile-specific storage options and features (for example, compress, encrypt, deduplicate) will cause an exception. The BasicFile LOB defaults will be used for storage options not specified.

- `PERMITTED`

  LOBs are allowed to be created as SecureFiles.

- `ALWAYS`

  All LOBs created in the system are created as SecureFile LOBs. If the LOB is not created in an Automatic Segment Space Managed tablespace, then an error will occur. Any BasicFile LOB storage options are ignored. The SecureFile defaults will be used for all storage options not specified.

- `IGNORE`

  The `SECUREFILE` keyword and all SecureFile options are ignored.

If the `COMPATIBLE` initialization parameter is not set to `11.1` or higher, then LOBs are not treated as SecureFiles.

If there is a LOB column with two partitions (one that has a tablespace for which ASSM is enabled and one that has a tablespace for which ASSM is not enabled), then LOBs in the partition with the ASSM-enabled tablespace will be treated as SecureFiles and LOBs in the other partition will be treated as BasicFile LOBs.

DB_ULTRA_SAFE

Property	Description
Parameter type	String
Syntax	`DB_ULTRA_SAFE = { OFF | DATA_ONLY | DATA_AND_INDEX }`

Property	Description
Default value	`OFF`
Modifiable	No
Basic	No

`DB_ULTRA_SAFE` sets the default values for other parameters that control protection levels.

Values:

- `OFF`

  When any of `DB_BLOCK_CHECKING`, `DB_BLOCK_CHECKSUM`, or `DB_LOST_WRITE_PROTECT` are explicitly set, no changes are made.

- `DATA_ONLY`
  - `DB_BLOCK_CHECKING` will be set to `MEDIUM`.
  - `DB_LOST_WRITE_PROTECT` will be set to `TYPICAL`.
  - `DB_BLOCK_CHECKSUM` will be set to `FULL`.

- `DATA_AND_INDEX`
  - `DB_BLOCK_CHECKING` will be set to `FULL`.
  - `DB_LOST_WRITE_PROTECT` will be set to `TYPICAL`.
  - `DB_BLOCK_CHECKSUM` will be set to `FULL`.

DB_UNIQUE_NAME

Property	Description
Parameter type	String
Syntax	`DB_UNIQUE_NAME =` *`database_unique_name`*
Default value	Database instances: the value of `DB_NAME` Automatic Storage Management instances: `+ASM`
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

`DB_UNIQUE_NAME` specifies a globally unique name for the database. Databases with the same `DB_NAME` within the same `DB_DOMAIN` (for example, copies of a database created for reporting or a physical standby) must have a unique `DB_UNIQUE_NAME`. Every database's `DB_UNIQUE_NAME` must be unique within the enterprise.

The value of `DB_UNIQUE_NAME` can be up to 30 characters and is case insensitive. The following characters are valid in a database name: alphanumeric characters, underscore (_), number sign (#), and dollar sign ($).

> **See Also:** *Oracle Data Guard Concepts and Administration* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DB_WRITER_PROCESSES

Property	Description
Parameter type	Integer
Default value	1 or `CPU_COUNT` / 8, whichever is greater
Modifiable	No
Range of values	1 to 20
Basic	No

`DB_WRITER_PROCESSES` is useful for systems that modify data heavily. It specifies the initial number of database writer processes for an instance.

See Also:

- *Oracle Database Concepts* for more information on the database writer processes
- *Oracle Database Performance Tuning Guide* for information on setting the DBW*n* parameters

DBWR_IO_SLAVES

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	No
Range of values	0 to operating system-dependent
Basic	No

`DBWR_IO_SLAVES` is relevant only on systems with only one database writer process (`DBW0`). It specifies the number of I/O server processes used by the `DBW0` process. The `DBW0` process and its server processes always write to disk. By default, the value is `0` and I/O server processes are not used.

If you set `DBWR_IO_SLAVES` to a nonzero value, the number of I/O server processes used by the `ARCH` and `LGWR` processes is set to `4`. However, the number of I/O server processes used by Recovery Manager is set to `4` only if asynchronous I/O is disabled (either your platform does not support asynchronous I/O or `disk_asynch_io` is set to `false`).

Typically, I/O server processes are used to simulate asynchronous I/O on platforms that do not support asynchronous I/O or that implement it inefficiently. However, you can use I/O server processes even when asynchronous I/O is being used. In that case the I/O server processes will use asynchronous I/O.

I/O server processes are also useful in database environments with very large I/O throughput, even if asynchronous I/O is enabled.

See Also: "BACKUP_TAPE_IO_SLAVES" on page 1-26

DDL_LOCK_TIMEOUT

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SESSION`
Range of values	0 to 1,000,000 (in seconds)
Basic	No

`DDL_LOCK_TIMEOUT` specifies a time limit for how long DDL statements will wait in a DML lock queue. The default value of zero indicates a status of `NOWAIT`. The maximum value of 1,000,000 seconds will result in the DDL statement waiting forever to acquire a DML lock.

If a lock is not acquired before the timeout period expires, then an error is returned.

DG_BROKER_CONFIG_FILE*n*

Property	Description
Parameter type	String
Syntax	`DG_BROKER_CONFIG_FILE[1	2] =` *`filename`*
Default value	Operating system-dependent
Modifiable	`ALTER SYSTEM`
Range of values	One filename
Basic	No

`DG_BROKER_CONFIG_FILE`*`n`* (where *`n`* = 1, 2) specifies the names for the Data Guard broker configuration files.

Every database that is part of a Data Guard broker configuration has two broker configuration files, which contain entries that describe the state and properties of the configuration (such as the sites and databases that are part of the configuration, the roles and properties of each of the databases, and the state of each of the elements of the configuration). Two files are provided so as to always maintain the last known good state of the configuration.

If `DG_BROKER_CONFIG_FILE`*`n`* is not explicitly defined, then it is set to an operating system-specific default value at instance startup. The parameter can only be altered when the DMON (Data Guard broker) process is not running.

See Also: *Oracle Data Guard Broker* for more information about setting this parameter

DG_BROKER_START

Property	Description
Parameter type	Boolean

Property	Description
Default value	`false`
Modifiable	`ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`DG_BROKER_START` enables Oracle to determine whether or not the Data Guard broker (DMON) process should be started. DMON is a non-fatal Oracle background process and exists as long as the instance exists, whenever this parameter is set to `true`.

If the site is never going to be configured in a Data Guard broker configuration, then you can leave the parameter unspecified and accept the default value of `false`. If the site is part of a Data Guard broker configuration, then automatic startup of the DMON process is simplified by setting `DG_BROKER_START` to `true` in the initialization parameter file.

DIAGNOSTIC_DEST

Property	Description
Parameter type	String
Syntax	`DIAGNOSTIC_DEST = { pathname | directory }`
Default value	Derived from the value of `ORACLE_BASE ($ORACLE_BASE)`. If the `$ORACLE_BASE` environment variable does not exist, then `$ORACLE_HOME` is used.
Modifiable	`ALTER SYSTEM`
Basic	No
Real Application Clusters	This parameter can be set on each instance. Oracle recommends that each instance in a cluster specify a `DIAGNOSTIC_DEST` directory location that is located on shared disk and that the same value for `DIAGNOSTIC_DEST` be specified for each instance.

As of Oracle Database 11g Release 1, the diagnostics for each database instance are located in a dedicated directory, which can be specified through the `DIAGNOSTIC_DEST` initialization parameter. The structure of the directory specified by `DIAGNOSTIC_DEST` is as follows:

```
<diagnostic_dest>/diag/rdbms/<dbname>/<instname>
```

This location is known as the Automatic Diagnostic Repository (ADR) Home. For example, if the database name is `proddb` and the instance name is `proddb1`, the ADR home directory would be `<diagnostic_dest>/diag/rdbms/proddb/proddb1`.

The following files are located under the ADR home directory:

- Trace files - located in subdirectory `<adr-home>/trace`
- Alert logs - located in subdirectory `<adr-home>/alert`. In addition, the `alert.log` file is now in XML format, which conforms to the Oracle ARB logging standard.
- Core files - located in the subdirectory `<adr-home>/cdumd`

- Incident files - the occurrence of each serious error (for example, ORA-600, ORA-1578, ORA-7445) causes an incident to be created. Each incident is assigned an ID and dumping for each incident (error stack, call stack, block dumps, and so on) is stored in its own file, separated from process trace files. Incident dump files are located in `<adr-home>/incident/<incdir#>`. You can find the incident dump file location inside the process trace file.

DISK_ASYNCH_IO

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	No
Range of values	`true` | `false`
Basic	No

`DISK_ASYNCH_IO` controls whether I/O to datafiles, control files, and logfiles is asynchronous (that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans). If your platform supports asynchronous I/O to disk, Oracle recommends that you leave this parameter set to its default value. However, if the asynchronous I/O implementation is not stable, you can set this parameter to `false` to disable asynchronous I/O. If your platform does not support asynchronous I/O to disk, this parameter has no effect.

If you set `DISK_ASYNCH_IO` to `false`, then you should also set `DBWR_IO_SLAVES` to a value other than its default of zero in order to simulate asynchronous I/O.

See Also:

- "DBWR_IO_SLAVES" on page 1-54
- *Oracle Database Performance Tuning Guide* for more information on asynchronous I/O and on setting this parameter

DISPATCHERS

Property	Description
Parameter type	String

Syntax	`DISPATCHERS = '`*`dispatch_clause`*`'`
	dispatch_clause::=
	`(PROTOCOL = `*`protocol`*`) |`
	`(ADDRESS = `*`address`*`) |`
	`(DESCRIPTION = `*`description`*` )`
	`[options_clause]`

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td></td><td><b>options_clause::=</b>
(DISPATCHERS = <i>integer</i> |
 SESSIONS = <i>integer</i> |
 CONNECTIONS = <i>integer</i> |
 TICKS = <i>seconds</i> |
 POOL = {1 | ON | YES | TRUE | BOTH |
   ({IN | OUT} = <i>ticks</i>) | 0 | OFF | NO | FALSE |
   <i>ticks</i>} |
 MULTIPLEX = {1 | ON | YES | TRUE |
   0 | OFF | NO | FALSE | BOTH | IN | OUT} |
 LISTENER = <i>tnsname</i> |
 SERVICE = <i>service</i> |
 INDEX = <i>integer</i>)</td></tr>
<tr><td><b>Default value</b></td><td>If SHARED_SERVERS is greater than 0, then DISPATCHERS defaults to '(PROTOCOL=tcp)', causing 1 TCP/IP dispatcher to be created.</td></tr>
<tr><td><b>Modifiable</b></td><td>ALTER SYSTEM</td></tr>
<tr><td><b>Basic</b></td><td>No</td></tr>
</table>

DISPATCHERS configures dispatcher processes in the shared server architecture. The parsing software supports a name-value syntax to enable the specification of attributes in a position-independent, case-insensitive manner. For example:

```
DISPATCHERS = '(PROTOCOL=TCP)(DISPATCHERS=3)'
```

Attributes may be specified using the full attribute name or any substring beginning with the first 3 characters. For example, SESSIONS can be specified as SES, SESS, SESSI, and so on.

Specify only one of the following attributes: PROTOCOL, ADDRESS, or DESCRIPTION. If you specify either ADDRESS or DESCRIPTION, then you can specify additional network attributes. Doing so supports multi-homed hosts.

dispatch_clause

- PROTOCOL

  The network protocol for which the dispatcher generates a listening endpoint.

- ADDRESS

  The network protocol address of the endpoint on which the dispatchers listen.

- DESCRIPTION

  The network description of the endpoint on which the dispatchers listen, including the protocol address.

options_clause

- DISPATCHERS

  The initial number of dispatchers to start. The default is 1.

- SESSIONS

The maximum number of network sessions to allow for each dispatcher. The default is operating system-specific. Most operating systems have a default of 16 KB.

- `CONNECTIONS`

  The maximum number of network connections to allow for each dispatcher. The default is operating system-specific.

- `TICKS`

  The length of a network tick in seconds. The default is 1 second.

- `POOL`

  Enables Connection Pooling.

  - An integer indicates that Connection Pooling is enabled for both incoming and outgoing network connections. The number specified is the timeout in ticks for both incoming and outgoing network connections.
  - `ON`, `YES`, `TRUE`, and `BOTH` indicate that Connection Pooling is enabled for both incoming and outgoing network connections. A default timeout of 10 ticks is used for both incoming and outgoing network connections.
  - `IN` indicates that Connection Pooling is enabled for incoming network connections and the default timeout of 10 ticks is used for incoming network connections.
  - `OUT` indicates that Connection Pooling is enabled for outgoing network connections and the default timeout of 10 ticks is used for outgoing network connections.
  - `NO`, `OFF`, and `FALSE` indicate that Connection Pooling is disabled for both incoming and outgoing network connections. This is the default.

  `POOL` can also be assigned a name-value string such as: "(IN=10)", "(OUT=20)", or "(IN=10)(OUT=20)". In such cases:

  - If an `IN` numeric value is specified, then Connection Pooling is enabled for incoming connections, and the number specified is the timeout in ticks for incoming network connections.
  - If an `OUT` numeric value is specified, then Connection Pooling is enabled for outgoing network connections, and the number specified is the timeout in ticks for outgoing network connections.
  - If the numeric value of a specified timeout is 1, then the default value of 10 ticks is used. If the numeric value is 0, then pooling is not on.

- `MULTIPLEX`

  Enables the Oracle Connection Manager session multiplexing feature.

  - The values `1`, `ON`, `YES`, `TRUE`, and `BOTH` indicate that Network Session Multiplex is enabled for both incoming and outgoing network connections.
  - The value `IN` indicates that Network Session Multiplex is enabled for incoming network connections.
  - The value `OUT` indicates that Network Session Multiplexing is enabled for outgoing network connections.
  - The values `0`, `NO`, `OFF`, and `FALSE` indicate that Network Session Multiplexing is disabled for both incoming and outgoing network connections. This is the default.

- LISTENER

  Specifies the network name of an address or address list of the Oracle Net listeners with which the dispatchers will register.

  The LISTENER attribute facilitates administration of multi-homed hosts. This attribute specifies the appropriate listeners with which the dispatchers will register. The LISTENER attribute takes precedence over the LOCAL_LISTENER and REMOTE_LISTENER parameters. See "LOCAL_LISTENER" on page 1-78 and "REMOTE_LISTENER" on page 1-133.

- SERVICE

  Specifies one or more names by which clients can connect to the dispatchers. The SERVICE attribute takes precedence over the SERVICE_NAMES parameter.

- INDEX

  Use this attribute in an ALTER SYSTEM SET DISPATCHERS statement to indicate which dispatcher configuration you want to modify. (If you specify INDEX in the initialization parameter file, the Oracle Database ignores it.) In an ALTER SYSTEM statement, INDEX specifies the order in which the parameter's values were initialized. The value ranges from 0 (for the first dispatcher configuration) to one less than the total number of dispatcher configurations you define.

  For example, if you specify 3 dispatcher configurations in the initialization parameter file, you would modify the third dispatcher configuration by specifying INDEX=2 in the ALTER SYSTEM statement. You could also add an additional dispatcher configuration in the ALTER SYSTEM statement by specifying INDEX=3.

  If INDEX is not specified in the ALTER SYSTEM statement, then the PROTOCOL, ADDRESS, or DESCRIPTION attributes must be specified, and if a dispatcher configuration matching this PROTOCOL, ADDRESS, or DESCRIPTION exists, then that configuration will be modified. Otherwise, a new configuration will be added.

> **See Also:** *Oracle Database Net Services Administrator's Guide* and *Oracle Database Administrator's Guide* for more information on setting this parameter

DISTRIBUTED_LOCK_TIMEOUT

Property	Description
Parameter type	Integer
Default value	60
Modifiable	No
Range of values	1 to unlimited
Basic	No

DISTRIBUTED_LOCK_TIMEOUT specifies the amount of time (in seconds) for distributed transactions to wait for locked resources.

> **See Also:** *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for more information on data concurrency

DML_LOCKS

Property	Description
Parameter type	Integer
Default value	Derived: 4 * `TRANSACTIONS`
Modifiable	No
Range of values	20 to unlimited; a setting of 0 disables enqueues
Basic	No
Real Application Clusters	You must set this parameter for every instance, and all instances must have positive values or all must be 0.

A **DML lock** is a lock obtained on a table that is undergoing a DML operation (insert, update, delete). `DML_LOCKS` specifies the maximum number of DML locks—one for each table modified in a transaction. The value should equal the grand total of locks on tables currently referenced by all users. For example, if three users are modifying data in one table, then three entries would be required. If three users are modifying data in two tables, then six entries would be required.

The default value assumes an average of four tables referenced for each transaction. For some systems, this value may not be enough.

Enqueues are shared memory structures that serialize access to database resources. If you set the value of `DML_LOCKS` to `0`, enqueues are disabled and performance is slightly increased. However, you should be aware of the following restrictions when you set you `DML_LOCKS` to `0`:

- You cannot use `DROP TABLE`, `CREATE INDEX` statements
- You cannot use explicit lock statements such as `LOCK TABLE IN EXCLUSIVE MODE`
- Enterprise Manager cannot run on any instances for which `DML_LOCKS` is set to `0`

Oracle holds more locks during parallel DML than during serial execution. Therefore, if your database supports a lot of parallel DML, you may need to increase the value of this parameter.

See Also:

- *Oracle Database Concepts* for a discussion of lock and enqueue resources needed for parallel DML
- *Oracle Real Application Clusters Administration and Deployment Guide*, *Oracle Database Concepts*, and *Oracle Database Administrator's Guide* for more information on data concurrency

EVENT

Property	Description
Parameter type	String
Default value	There is no default value.
Modifiable	No
Basic	No

EVENT is a parameter used only to debug the system. Do not alter the value of this parameter except under the supervision of Oracle Support Services staff.

FAL_CLIENT

Property	Description
Parameter type	String
Syntax	`FAL_CLIENT = string`
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

FAL_CLIENT specifies the FAL (fetch archive log) client name that is used by the FAL service, configured through the FAL_SERVER parameter, to refer to the FAL client. The value is an Oracle Net service name, which is assumed to be configured properly on the FAL server system to point to the FAL client (standby database).

Given the dependency of FAL_CLIENT on FAL_SERVER, the two parameters should be configured or changed at the same time.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about FAL server

FAL_SERVER

Property	Description
Parameter type	String
Syntax	`FAL_SERVER = string`
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Basic	No

FAL_SERVER specifies the FAL (fetch archive log) server for a standby database. The value is an Oracle Net service name, which is assumed to be configured properly on the standby database system to point to the desired FAL server.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about FAL server

FAST_START_MTTR_TARGET

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM

Property	Description
Range of values	0 to 3600 seconds
Basic	No
Real Application Clusters	Multiple instances can have different values, and you can change the values at runtime.

FAST_START_MTTR_TARGET enables you to specify the number of seconds the database takes to perform crash recovery of a single instance. When specified, FAST_START_MTTR_TARGET is overridden by LOG_CHECKPOINT_INTERVAL.

FAST_START_PARALLEL_ROLLBACK

Property	Description
Parameter type	String
Syntax	FAST_START_PARALLEL_ROLLBACK = { HI | LO | FALSE }
Default value	LOW
Modifiable	ALTER SYSTEM
Basic	No

FAST_START_PARALLEL_ROLLBACK specifies the degree of parallelism used when recovering terminated transactions. Terminated transactions are transactions that are active before a system failure. If a system fails when there are uncommitted parallel DML or DDL transactions, then you can speed up transaction recovery during startup by using this parameter.

Values:

- FALSE

  Parallel rollback is disabled

- LOW

  Limits the maximum degree of parallelism to 2 * CPU_COUNT

- HIGH

  Limits the maximum degree of parallelism to 4 * CPU_COUNT

If you change the value of this parameter, then transaction recovery will be stopped and restarted with the new implied degree of parallelisme.

FILE_MAPPING

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Range of values	true | false
Basic	No

`FILE_MAPPING` enables or disables file mapping. The FMON background process will be started to manage the mapping information when file mapping is enabled.

FILEIO_NETWORK_ADAPTERS

Property	Description
Parameter type	String
Syntax	`FILEIO_NETWORK_ADAPTERS =` *adapter_name* `[,` *adapter_name* `] ...`
Default value	There is no default value.
Modifiable	No
Range of values	One or more network adapter names, separated by commas
Basic	No

`FILEIO_NETWORK_ADAPTERS` specifies a list of network adapters that can be used to access the disk storage. On platforms where the database files reside in network attached storage, this parameter provides the storage access library the list of network adapters that can be used to access the storage.

The netrwork adapter name is a fully qualified address name of the network card that can be accessed through the host name database or using the Network Information Service. The components of the adapter name are separated by periods. For example, the following is a fully qualified adapter name:

```
ib1.oracle.com
```

FILESYSTEMIO_OPTIONS

Property	Description
Parameter type	String
Syntax	`FILESYSTEMIO_OPTIONS = { none | setall | directIO | asynch }`
Default value	There is no default value.
Modifiable	No
Basic	No

`FILESYSTEMIO_OPTIONS` specifies I/O operations for file system files.

FIXED_DATE

Property	Description
Parameter type	String
Syntax	`FIXED_DATE = [YYYY-MM-DD-HH24:MI:SS` (or the default Oracle date format) | `NONE]`
Default value	There is no default value.

Property	Description
Modifiable	`ALTER SYSTEM`
Basic	No

`FIXED_DATE` enables you to set a constant date that `SYSDATE` will always return instead of the current date. To undo a fixed date setting, specify `FIXED_DATE=NONE`. This parameter is useful primarily for testing. The value can be in the format shown above or in the default Oracle date format, without a time.

GC_FILES_TO_LOCKS

Property	Description
Parameter type	String

Syntax	`GC_FILES_TO_LOCKS =`
	`'{`*`file_list`*`=`*`lock_count`*`[!`*`blocks`*`][EACH][:...]}'`
	Spaces are not allowed within the quotation marks.

Default value	There is no default value.
Modifiable	No
Basic	No
Real Application Clusters	You must set this parameter for every instance, and multiple instances must have identical values. To change the value, you must shut down all instances in the cluster, change the value for each instance, and then start up each instance.

> **Note:** Setting this parameter to any value other than the default will disable Cache Fusion processing in a Real Application Clusters environment.

`GC_FILES_TO_LOCKS` is a Real Application Clusters parameter that has no effect on an instance running in exclusive mode. It controls the mapping of pre-release 9.0.1 parallel cache management (PCM) locks to datafiles.

Values:

- *`file_list`*

  One or more datafiles listed by their file numbers, or ranges of file numbers, with comma separators:

  ```
  filenumber[-filenumber][,filenumber[-filenumber]]...
  ```

  To find the correspondence between filenames and file numbers, query the `FILE_NAME` and `FILE_ID` columns of the `DBA_DATA_FILES` data dictionary view.

- *`lock_count`*

  The number of PCM locks assigned to *`file_list`*. By default these locks are fixed. If you set *`lock_count`* to `0`, then Oracle uses fine-grain locking for these files and takes locks as needed from the pool of releasable locks.

- *`blocks`*

Specifies the number of contiguous blocks covered by one lock. The default is noncontiguous blocks.

- EACH

  Indicates that each datafile in *file_list* is assigned a separate set of *lock_count* PCM locks.

The value of the parameter should be set to cover as many files as possible. Therefore, to avoid performance problems, you should always change GC_FILES_TO_LOCKS when the size of datafiles change or when new datafiles are added. Doing so requires you to shut down and restart your cluster database.

If the number of PCM locks allocated to a datafile is less than or equal to the number of blocks in a datafile, each of these locks will cover a number of contiguous blocks within the datafile equal to *blocks*. If the number of PCM locks assigned to the datafile is larger than its number of blocks, resources will be wasted because some locks will not be covering any blocks.

A colon (:) separates each clause that assigns a number of PCM locks to *file_list*.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* and the *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system for more information on setting this parameter

GCS_SERVER_PROCESSES

Property	Description
Parameter type	Integer
Default value	For one CPU, there will be one GCS server process. For 2 - 8 CPUs, there will be 2 GCS server processes. For more than 8 CPUs, the number of GCS server processes will be equal to the number of CPUs divided by 4. If the result includes a fraction, the fraction is disregarded. For example, if you had 10 CPUs, then 10/4 would mean 2 GCS processes.
Modifiable	No
Range of values	1 to 36
Basic	No
Real Application Clusters	Multiple instances can have different values.

GCS_SERVER_PROCESSES specifies the number of server processes in Global Cache Service (GCS) to serve the inter-instance traffic among Real Application Clusters (RAC) instances. GCS server processes are only seen in a RAC environment.

GLOBAL_CONTEXT_POOL_SIZE

Property	Description
Parameter type	String
Default value	1 MB

Property	Description
Modifiable	No
Range of values	Any integer value in MB
Basic	No

Note: The GLOBAL_CONTEXT_POOL_SIZE parameter is deprecated. It is retained for backward compatibility only.

GLOBAL_CONTEXT_POOL_SIZE specifies the amount of memory to allocate in the SGA for storing and managing global application context.

GLOBAL_NAMES

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true | false
Basic	No

GLOBAL_NAMES specifies whether a database link is required to have the same name as the database to which it connects.

If the value of GLOBAL_NAMES is false, then no check is performed. If you use or plan to use distributed processing, then Oracle recommends that you set this parameter to true to ensure the use of consistent naming conventions for databases and links in a networked environment.

See Also: *Oracle Database Administrator's Guide* for more information on setting this parameter

GLOBAL_TXN_PROCESSES

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SYSTEM
Range of values	1 to 20
Basic	No
Real Application Clusters	Multiple instances can have different values.

GLOBAL_TXN_PROCESSES specifies the initial number of GTX*n* background processes (GTX0, ... GTX9 and GTXa, ... GTXj) per instance to support global (XA) transactions in an Oracle RAC environment.

GLOBAL_TXN_PROCESSES is useful for systems that process global (XA) transactions heavily. You do not need to specify a value for this parameter since Oracle Database automatically determines the number of processes and autotunes them, as necessary. GTX*n* background processes are only seen in an Oracle RAC environment.

HASH_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	Derived: 2 * SORT_AREA_SIZE
Modifiable	ALTER SESSION
Range of values	0 to operating system-dependent
Basic	No

Note: Oracle does not recommend using the HASH_AREA_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. HASH_AREA_SIZE is retained for backward compatibility.

HASH_AREA_SIZE is relevant to parallel execution operations and to the query portion of DML or DDL statements. It specifies the maximum amount of memory, in bytes, to be used for hash joins.

See Also:

- *Oracle Database Concepts* for information on hash joins in general
- *Oracle Database Performance Tuning Guide* for information on calculating an appropriate value for this parameter

HI_SHARED_MEMORY_ADDRESS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Basic	No

HI_SHARED_MEMORY_ADDRESS specifies the starting address at runtime of the system global area (SGA). It is ignored on platforms that specify the SGA's starting address at linktime.

On 64-bit platforms, use HI_SHARED_MEMORY_ADDRESS to specify the high-order 32 bits of a 64-bit address. Use SHARED_MEMORY_ADDRESS to specify the low-order 32 bits of the address (see "SHARED_MEMORY_ADDRESS" on page 1-147). If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

HS_AUTOREGISTER

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SYSTEM`
Range of values	`true` | `false`
Basic	No

`HS_AUTOREGISTER` enables or disables automatic self-registration of Heterogeneous Services (HS) agents. When enabled, information is uploaded into the server's data dictionary to describe a previously unknown agent class or a new agent version.

Oracle recommends that you set this parameter to `true`. Oracle incurs less overhead when establishing subsequent connections through the same agent if self-registered information is available in the server's data dictionary.

> **See Also:** *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information on HS agents

IFILE

Property	Description
Parameter type	Parameter file
Syntax	`IFILE =` *`parameter_file_name`*
Default value	There is no default value.
Modifiable	No
Range of values	Valid parameter filenames
Basic	No
Real Application Clusters	Multiple instances can have different values.

Use `IFILE` to embed another parameter file within the current parameter file. For example:

```
IFILE = COMMON.ORA
```

You can have up to three levels of nesting. In this example, the file `COMMON.ORA` could contain a second `IFILE` parameter for the file `COMMON2.ORA`, which could contain a third `IFILE` parameter for the file `GCPARMS.ORA`. You can also include multiple parameter files in one parameter file by listing `IFILE` several times with different values:

```
IFILE = DBPARMS.ORA
IFILE = GCPARMS.ORA
IFILE = LOGPARMS.ORA
```

> **Note:** You must list multiple entries on contiguous lines of the parameter file.

INSTANCE_GROUPS

Property	Description
Parameter type	String
Syntax	`INSTANCE_GROUPS =` *`group_name`* `[,` *`group_name`* `] ...`
Default value	There is no default value.
Modifiable	No
Range of values	One or more instance group names, separated by commas
Basic	No
Real Application Clusters	Multiple instances can have different values.

> **Note:** The `INSTANCE_GROUPS` parameter is deprecated. It is retained for backward compatibility only.

`INSTANCE_GROUPS` is a Real Application Clusters parameter that you can specify only in parallel mode. Used in conjunction with the `PARALLEL_INSTANCE_GROUP` parameter, it lets you restrict parallel query operations to a limited number of instances.

This parameter specifies one or more instance groups and assigns the current instance to those groups. If one of the specified groups is also specified in the `PARALLEL_INSTANCE_GROUP` parameter, then Oracle allocates query processes for a parallel operation from this instance.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information on parallel query execution in a Real Application Clusters environment
- "PARALLEL_INSTANCE_GROUP" on page 1-118

INSTANCE_NAME

Property	Description
Parameter type	String
Syntax	`INSTANCE_NAME =` *`instance_id`*
Default value	The instance's SID **Note:** The SID identifies the instance's shared memory on a host, but may not uniquely distinguish this instance from other instances.

Property	Description
Modifiable	No
Range of values	Any alphanumeric characters
Basic	No

In a Real Application Clusters environment, multiple instances can be associated with a single database service. Clients can override Oracle's connection load balancing by specifying a particular instance by which to connect to the database. `INSTANCE_NAME` specifies the unique name of this instance.

In a single-instance database system, the instance name is usually the same as the database name.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* and *Oracle Database Net Services Administrator's Guide* for more information

INSTANCE_NUMBER

Property	Description
Parameter type	Integer
Default value	0 (zero)
Modifiable	No
Range of values	1 to maximum number of instances specified when the database was created
Basic	Yes
Real Application Clusters	You must set this parameter for every instance, and all instances must have different values.

`INSTANCE_NUMBER` is a Real Application Clusters parameter that can be specified in parallel mode or exclusive mode. It specifies a unique number that maps the instance to one free list group for each database object created with storage parameter `FREELIST GROUPS`.

The `INSTANCE` parameter of the `ALTER TABLE ... ALLOCATE EXTENT` statement assigns an extent to a particular free list group. If you set `INSTANCE_NUMBER` to the value specified for the `INSTANCE` parameter, the instance uses that extent for inserts and for updates that expand rows.

The practical maximum value of this parameter is the maximum number of instances specified in the `CREATE DATABASE` statement. The absolute maximum is operating system-dependent.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for more information

INSTANCE_TYPE

Property	Description
Parameter type	String
Syntax	`INSTANCE_TYPE = { RDBMS | ASM }`
Default value	`RDBMS`
Modifiable	No
Basic	No
Real Application Clusters	Multiple instances must have the same value.

`INSTANCE_TYPE` specifies whether the instance is a database instance or an Automatic Storage Management instance.

Values:

- `RDBMS`

  The instance is a database instance.

- `ASM`

  The instance is an Automatic Storage Management instance.

JAVA_JIT_ENABLED

Property	Description
Parameter type	Boolean
Default value	Operating system-dependent
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`JAVA_JIT_ENABLED` enables or disables the Just-in-Time (JIT) compiler for the Oracle Java Virtual Machine (OracleJVM) environment.

For platforms that support the JIT compiler, the default value of this parameter is `true`; otherwise the default value is `false`. Attempting to set this parameter to `true` on unsupported platforms will result in an error.

> **See Also:** *Oracle Database Java Developer's Guide* for more information about setting this parameter

JAVA_MAX_SESSIONSPACE_SIZE

Property	Description
Parameter type	Integer
Default value	`0`

Property	Description
Modifiable	No
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. `JAVA_MAX_SESSIONSPACE_SIZE` specifies (in bytes) the maximum amount of session space made available to a Java program executing in the server. When a user's session-duration Java state attempts to exceed this amount, the Java virtual machine kills the session with an out-of-memory failure.

See Also:

- *Oracle Database Java Developer's Guide*
- "JAVA_SOFT_SESSIONSPACE_LIMIT" on page 1-73

JAVA_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	`JAVA_POOL_SIZE =` *`integer`* `[K | M | G]`
Default value	If `SGA_TARGET` is set: If the parameter is not specified, then the default is `0` (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If `SGA_TARGET` is not set: 24 MB, rounded up to the nearest granule size
Modifiable	`ALTER SYSTEM`
Range of values	Minimum: `0` (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

`JAVA_POOL_SIZE` specifies (in bytes) the size of the Java pool, from which the Java memory manager allocates most Java state during runtime execution. This memory includes the shared in-memory representation of Java method and class definitions, as well as the Java objects that are migrated to the Java session space at end-of-call.

See Also: *Oracle Database Java Developer's Guide* for information on adjusting this parameter

JAVA_SOFT_SESSIONSPACE_LIMIT

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	No

Property	Description
Range of values	0 to 2 GB - 1
Basic	No

Java session space is the memory that holds Java state from one database call to another. `JAVA_SOFT_SESSIONSPACE_LIMIT` specifies (in bytes) a **soft limit** on Java memory usage in a session, as a means to warn you if a user's session-duration Java state is using too much memory. When a user's session-duration Java state exceeds this size, Oracle generates a warning that goes into the trace files.

See Also:

- *Oracle Database Java Developer's Guide* for more information on this parameter
- "JAVA_MAX_SESSIONSPACE_SIZE" on page 1-72

JOB_QUEUE_PROCESSES

Property	Description
Parameter type	Integer
Default value	`1000`
Modifiable	`ALTER SYSTEM`
Range of values	0 to 1000
Basic	Yes
Real Application Clusters	Multiple instances can have different values.

`JOB_QUEUE_PROCESSES` specifies the maximum number of processes that can be created for the execution of jobs. It specifies the number of job queue processes per instance (J000, ... J999). Replication uses job queues for data refreshes. Advanced queuing uses job queues for message propagation. You can create user job requests through the `DBMS_JOB` package.

Some job queue requests are created automatically. An example is refresh support for materialized views. If you wish to have your materialized views updated automatically, you must set `JOB_QUEUE_PROCESSES` to a value of one or higher.

See Also:

- *Oracle Database Advanced Replication* and *Oracle Database Data Warehousing Guide* for more information on managing materialized views
- *Oracle Streams Advanced Queuing User's Guide* for more information on message propagation

Note: The `JOB_QUEUE_PROCESSES` parameter is deprecated. It is retained for backward compatibility only.

LARGE_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	`LARGE_POOL_SIZE = integer [K | M | G]`
Default value	If `SGA_TARGET` is set, but a value is not specified for `LARGE_POOL_SIZE`, then the default is 0 (internally determined by the Oracle database). If `LARGE_POOL_SIZE` is specified, then the user-specified value indicates a minimum value for the memory pool. If `SGA_TARGET` is not set, 0 if both of the following are true: ■ The pool is not required by parallel execution ■ `DBWR_IO_SLAVES` is not set Otherwise, derived from the values of `PARALLEL_MAX_SERVERS`, `PARALLEL_THREADS_PER_CPU`, `CLUSTER_DATABASE_INSTANCES`, `DISPATCHERS`, and `DBWR_IO_SLAVES`. Note that a value derived in this way does not take into account the requirements used for Automatic Storage Management files. As a general guideline, you should add 600K to the size of the SGA on a database instance using ASM.
Modifiable	`ALTER SYSTEM`
Range of values	300 KB to at least 2 GB (actual maximum is operating system-specific)
Basic	No

`LARGE_POOL_SIZE` specifies (in bytes) the size of the large pool allocation heap. The large pool allocation heap is used in shared server systems for session memory, by parallel execution for message buffers, and by backup processes for disk I/O buffers. Parallel execution allocates buffers out of the large pool only when `SGA_TARGET` is set.

Caution: When Oracle derives a default value, it adds 250K for each session for the shared server if `DISPATCHERS` is configured. The final derived value also includes a port-specific amount of memory for backup I/O buffers. The total derived default value can either be too large to allocate or can cause performance problems. In that case, set `LARGE_POOL_SIZE` to a number sufficiently small so that the database can start.

You can specify the value of this parameter using a number, optionally followed by K or M to specify kilobytes or megabytes, respectively. If you do not specify K or M, then the number is taken as bytes.

See Also: *Oracle Database Performance Tuning Guide* and *Oracle Database Upgrade Guide* for more information on setting this parameter

LDAP_DIRECTORY_ACCESS

Property	Description
Parameter type	String
Syntax	`LDAP_DIRECTORY_ACCESS = { NONE | PASSWORD | SSL }`

Property	Description
Default value	NONE
Modifiable	ALTER SYSTEM
Basic	No

LDAP_DIRECTORY_ACCESS specifies whether Oracle refers to Oracle Internet Directory for user authentication information. If directory access is turned on, then this parameter also specifies how users are authenticated.

Values:

- NONE

  Oracle does not refer to Oracle Internet Directory for Enterprise User Security information.

- PASSWORD

  Oracle tries to connect to the enterprise directory service using the database password stored in the database wallet. If that fails, then the Oracle Internet Directory connection fails and the database will not be able to retrieve enterprise roles and schema mappings upon enterprise user login.

- SSL

  Oracle tries to connect to Oracle Internet Directory using SSL.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide* for more information on Enterprise User Security

LDAP_DIRECTORY_SYSAUTH

Property	Description
Parameter type	String
Syntax	LDAP_DIRECTORY_SYSAUTH = { yes | no }
Default value	no
Modifiable	No
Basic	Yes

LDAP_DIRECTORY_SYSAUTH enables or disables directory-based authorization for SYSDBA and SYSOPER.

LICENSE_MAX_SESSIONS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to number of session licenses

Property	Description
Basic	No
Real Application Clusters	Multiple instances can have different values, but the total for all instances mounting a database should be less than or equal to the total number of sessions licensed for that database.

`LICENSE_MAX_SESSIONS` specifies the maximum number of concurrent user sessions allowed. When this limit is reached, only users with the `RESTRICTED SESSION` privilege can connect to the database. Users who are not able to connect receive a warning message indicating that the system has reached maximum capacity.

A zero value indicates that concurrent usage (session) licensing is not enforced. If you set this parameter to a nonzero number, you might also want to set `LICENSE_SESSIONS_WARNING` (see "LICENSE_SESSIONS_WARNING" on page 1-77).

Do not enable both concurrent usage licensing and user licensing. Set either `LICENSE_MAX_SESSIONS` or `LICENSE_MAX_USERS` to zero.

> **See Also:** *Oracle Database Administrator's Guide* for more information on setting this parameter

LICENSE_MAX_USERS

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SYSTEM`
Range of values	0 to number of user licenses
Basic	No
Real Application Clusters	Oracle recommends that multiple instances have the same value.

`LICENSE_MAX_USERS` specifies the maximum number of users you can create in the database. When you reach this limit, you cannot create more users. You can, however, increase the limit.

Do not enable both concurrent usage (session) licensing and user licensing. Set either `LICENSE_MAX_SESSIONS` or `LICENSE_MAX_USERS` to zero.

> **See Also:** *Oracle Database Administrator's Guide* for more information on setting this parameter

LICENSE_SESSIONS_WARNING

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SYSTEM`
Range of values	0 to value of `LICENSE_MAX_SESSIONS` parameter

Property	Description
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LICENSE_SESSIONS_WARNING` specifies a warning limit on the number of concurrent user sessions. When this limit is reached, additional users can connect, but Oracle writes a message in the alert file for each new connection. Users with `RESTRICTED SESSION` privilege who connect after the limit is reached receive a warning message stating that the system is nearing its maximum capacity.

If this parameter is set to zero, no warning is given as you approach the concurrent usage (session) limit. If you set this parameter to a nonzero number, you should also set `LICENSE_MAX_SESSIONS` (see "LICENSE_MAX_SESSIONS" on page 1-76).

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information on calculating an appropriate value for this parameter

LOCAL_LISTENER

Property	Description
Parameter type	String
Syntax	`LOCAL_LISTENER = `*`network_name`*
Default value	`(ADDRESS = (PROTOCOL=TCP)(HOST=`*`hostname`*`)(PORT=1521))` where *`hostname`* is the network name of the local host.
Modifiable	`ALTER SYSTEM`
Basic	No

`LOCAL_LISTENER` specifies a network name that resolves to an address or address list of Oracle Net local listeners (that is, listeners that are running on the same machine as this instance). The address or address list is specified in the `TNSNAMES.ORA` file or other address repository as configured for your system.

See Also:

- *Oracle Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle Database Net Services Administrator's Guide* and your operating system-specific Oracle documentation for more information about specifying network addresses for the protocols on your system

LOCK_NAME_SPACE

Property	Description
Parameter type	String
Syntax	`LOCK_NAME_SPACE =` *namespace*
Default value	There is no default value.
Modifiable	No
Range of values	Up to 8 alphanumeric characters. No special characters allowed.
Basic	No

Note: The `LOCK_NAME_SPACE` parameter is deprecated. It is retained for backward compatibility only.

`LOCK_NAME_SPACE` specifies the namespace that the distributed lock manager (DLM) uses to generate lock names. Consider setting this parameter if a standby or clone database has the same database name on the same cluster as the primary database.

If the standby database resides on the same file system as the primary database, then set `LOCK_NAME_SPACE` in the standby parameter file to a distinct value such as the following:

```
LOCK_NAME_SPACE = standby
```

LOCK_SGA

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`
Basic	No

`LOCK_SGA` locks the entire SGA into physical memory. It is usually advisable to lock the SGA into real (physical) memory, especially if the use of virtual memory would include storing some of the SGA using disk space. This parameter is ignored on platforms that do not support it.

LOG_ARCHIVE_CONFIG

Property	Description
Parameter type	String

Property	Description

Syntax	`LOG_ARCHIVE_CONFIG =`
	`{`
	`[ SEND │ NOSEND ]`
	`[ RECEIVE │ NORECEIVE ]`
	`[ DG_CONFIG=(`*remote_db_unique_name1* `[, ...` *remote_db_unique_name9*`) │ NODG_CONFIG ]`
	`}`

Default value	`'SEND, RECEIVE, NODG_CONFIG'`
Modifiable	`ALTER SYSTEM`
Basic	No

`LOG_ARCHIVE_CONFIG` enables or disables the sending of redo logs to remote destinations and the receipt of remote redo logs, and specifies the unique database names (`DB_UNIQUE_NAME`) for each database in the Data Guard configuration.

Values:

- `SEND`

  Enables the sending of redo logs to remote destinations

- `NOSEND`

  Disables the sending of redo logs to remote destinations

- `RECEIVE`

  Enables the receipt of remotely archived redo logs

- `NORECEIVE`

  Disables the receipt of remotely archived redo logs

- `DG_CONFIG`

  Specifies a list of up to 9 unique database names (defined with the `DB_UNIQUE_NAME` initialization parameter) for all of the databases in the Data Guard configuration.

- `NODG_CONFIG`

  Eliminates the list of service provider names previously specified with the `DG_CONFIG` option.

LOG_ARCHIVE_DEST

Property	Description
Parameter type	String
Syntax	`LOG_ARCHIVE_DEST =` *filespec*
Default value	Null
Modifiable	`ALTER SYSTEM`
Range of values	Any valid path or device name, except raw partitions
Basic	No

Property	Description
Real Application Clusters	Multiple instances can have different values.

> **Note:** For Enterprise Edition users, this parameter has been deprecated in favor of the LOG_ARCHIVE_DEST_*n* parameters. If Oracle Enterprise Edition is not installed or it is installed, but you have not specified any LOG_ARCHIVE_DEST_*n* parameters, this parameter is valid.

LOG_ARCHIVE_DEST is applicable only if you are running the database in ARCHIVELOG mode or are recovering a database from archived redo logs. LOG_ARCHIVE_DEST is incompatible with the LOG_ARCHIVE_DEST_*n* parameters, and must be defined as the null string ("") or (' ') when any LOG_ARCHIVE_DEST_*n* parameter has a value other than a null string. Use a text string to specify the default location and root of the disk file or tape device when archiving redo log files. (Archiving to tape is not supported on all operating systems.) The value cannot be a raw partition.

If LOG_ARCHIVE_DEST is not explicitly defined and all the LOG_ARCHIVE_DEST_*n* parameters have null string values, LOG_ARCHIVE_DEST is set to an operating system-specific default value on instance startup.

To override the destination that this parameter specifies, either specify a different destination for manual archiving or use the SQL*Plus statement ARCHIVE LOG START *filespec* for automatic archiving, where *filespec* is the new archive destination. To permanently change the destination, use the statement ALTER SYSTEM SET LOG_ARCHIVE_DEST = *filespec*, where *filespec* is the new archive destination.

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

See Also:

- *Oracle Database Backup and Recovery User's Guide*
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-86, "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-89, and "V$ARCHIVE_DEST" on page 6-8 for more information on setting this parameter
- Your Oracle operating system-specific documentation for the default value and for an example of how to specify the destination path or filename using LOG_ARCHIVE_DEST

LOG_ARCHIVE_DEST_*n*

Property	Description
Parameter type	String

Property	Description
Syntax	LOG_ARCHIVE_DEST_[1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10] = { *null_string* | { LOCATION=*path_name* | SERVICE=*service_name* } [MANDATORY] [REOPEN[=*seconds*]] [DELAY[=*minutes*]] [NOREGISTER] [ALTERNATE=*destination*] [DEPENDENCY=*destination*] [MAX_FAILURE=*count*] [SYNC | ASYNC] [AFFIRM | NOAFFIRM] [NET_TIMEOUT=*seconds*] [VALID_FOR=(*redo_log_type*,*database_role*)] [DB_UNIQUE_NAME] [MAX_CONNECTIONS=*count*] [COMPRESSION={ENABLE|DISABLE}] }
Default value	There is no default value.
Modifiable	Most attributes are modifiable by ALTER SESSION and ALTER SYSTEM, but some are not. See *Oracle Data Guard Concepts and Administration* for further information.
Basic	Yes

The LOG_ARCHIVE_DEST_*n* initialization parameter defines up to ten (where *n* = 1, 2, 3, ... 10) destinations, each of which *must* specify either the LOCATION or the SERVICE attribute to specify where to archive the redo data. All other attributes are optional. Note that whether you are specifying the LOCATION attribute or the SERVICE attribute, it must be the first attribute supplied in the list of attributes.

If you choose not to enter any attributes, you can specify a null string by entering the following:

```
LOG_ARCHIVE_DEST_n=' ';
```

You set the attributes for the LOG_ARCHIVE_DEST_*n* initialization parameter to control different aspects of how redo transport services transfer redo data from a production or primary database destination to another (standby) database destination. You can query the V$ARCHIVE_DEST view to see the current attribute settings for each destination (n).

Note: Some of the attributes for this parameter are deprecated, but are being retained for backward compatibility. See "Deprecated Attributes for LOG_ARCHIVE_DEST_n" on page 1-84.

For every LOG_ARCHIVE_DEST_n initialization parameter that you define, you must specify a corresponding LOG_ARCHIVE_DEST_STATE_n parameter. The LOG_ARCHIVE_DEST_STATE_n (where n is an integer from 1 to 10) initialization parameter specifies whether the corresponding destination is currently enabled or disabled.

See Also: "LOG_ARCHIVE_DEST_STATE_n" on page 1-85

Values:

> **See Also:** *Oracle Data Guard Concepts and Administration* for detailed descriptions of all the values listed in this section

- AFFIRM and NOAFFIRM

  Control whether or not redo transport services use synchronous or asynchronous disk I/O to write redo data to the archived redo log files. The default is NOAFFIRM.

- ALTERNATE=LOG_ARCHIVE_DEST_*n*

  Specifies an alternate archiving destination to be used when the original destination fails. There is no default value; if an alternate destination is not specified, then archiving does not automatically change to another destination if the original destination fails.

- ASYNC

  The redo data generated by a transaction need not have been received at a destination which has this attribute before that transaction can commit. This is the default behavior if neither SYNC nor ASYNC is specified.

- COMPRESSION

  Indicates whether network compression is enabled or disabled.

- DB_UNIQUE_NAME=*name*

  Specifies a unique name for the database at this destination. You must specify a name; there is no default value.

- DELAY[=*minutes*]

  Specifies a time lag between when redo data is archived on a physical standby site and when the archived redo log file is applied to the physical standby database. You cannot set the DELAY attribute for a logical standby database destination. If you specify the DELAY attribute without a time interval, the default is 30 minutes.

- DEPENDENCY=LOG_ARCHIVE_DEST_*n*

  Defines one archival destination to receive redo data on behalf of several destinations. Redo transport services transmit redo data to the destination that then shares its archived redo log files with multiple standby databases.

- LOCATION=*local_disk_directory* or USE_DB_RECOVERY_FILE_DEST

  Specifies either a local file system destination or the directory, file system, or Oracle Storage Manager disk group that will serve as the flash recovery area. You must specify this attribute for at least one destination. You can specify either a local disk directory or flash recovery area with the LOCATION attribute. You *must* include either the LOCATION or the SERVICE attribute for each destination to specify where to archive the redo data.

- MANDATORY

  Specifies that the transmission of redo data to the destination must succeed before the local online redo log file can be made available for reuse. If the MANDATORY attribute is not specified, then the destination is optional.

- MAX_CONNECTIONS

  Specifies the maximum number of network connections that can be used to transmit redo data to this destination. The default is 1.

- `MAX_FAILURE`

  Controls the consecutive number of times redo transport services attempt to reestablish communication and transmit redo data to a failed destination before the primary database permanently gives up on the standby database.

- `NET_TIMEOUT=`*seconds*

  Specifies the number of seconds the log writer process on the primary system waits for status from the network server (LNS*n*) process before terminating the network connection. The default is 30 seconds.

- `NOREGISTER`

  Indicates that the location of the archived redo log file is not to be recorded at the corresponding destination.

- `REOPEN`[=*seconds*]

  Specifies the minimum number of seconds before the archiver processes (ARC*n*) or the log writer process (LGWR) should try again to access a previously failed destination. The default is 300 seconds.

- `SERVICE=`*net_service_name*

  Specifies a valid Oracle Net service name (`SERVICE=`*net_service_name*) that identifies the remote Oracle database instance to which redo data will be sent. Each destination *must* specify either the `LOCATION` or the `SERVICE` attribute. There is no default net service name.

- `SYNC`

  The redo data generated by a transaction must have been received by every enabled destination which has this attribute before that transaction can commit.

- `VALID_FOR=(redo_log_type, database_role)`

  Identifies when redo transport services can transmit redo data to destinations based on the following factors:

  - `redo_log_type`—whether online redo log files, standby redo log files, or both are *currently* being archived on the database at this destination
  - `database_role`—whether the database is *currently* running in the primary or the standby role

Deprecated Attributes for LOG_ARCHIVE_DEST_n

The following attributes are deprecated for the `LOG_ARCHIVE_DEST_`*n* parameter. They are retained for backward compatibility only.

Table 1–2 Deprecated Attributes on the LOG_ARCHIVE_DEST_n Initialization Parameter

Deprecated Attribute	Alternative
`ASYNC`	Specify `SYNC` or `ASYNC`. `ASYNC` is the default if neither attribute is specified.
`LGWR`	Specify `SYNC` or `ASYNC`. `ASYNC` is the default if neither attribute is specified.
`OPTIONAL`	Destinations are optional by default.
`TEMPLATE`	Use `LOG_ARCHIVE_FORMAT`.
`VERIFY`	None. This attribute is only used with the deprecated `ARCH` attribute.

In addition, note the following changes to the ASYNC and SYNC attributes:

- The BLOCKS keyword on the ASYNC attribute is no longer needed.

  It is no longer necessary to set this keyword because Data Guard dynamically adjusts the block count up or down to an appropriate number of blocks, as necessary.

- The PARALLEL and NOPARALLEL keywords on the SYNC attribute are no longer needed.

LOG_ARCHIVE_DEST_STATE_*n*

Property	Description
Parameter type	String
Syntax	`LOG_ARCHIVE_DEST_STATE_`*n* `= { alternate | reset | defer | enable }`
Default value	`enable`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	Yes

The LOG_ARCHIVE_DEST_STATE_*n* parameters (where *n* = 1, 2, 3, ... 10) specify the availability state of the corresponding destination. The parameter suffix (1 through 10) specifies one of the ten corresponding LOG_ARCHIVE_DEST_*n* destination parameters.

Values:

- `enabled`

  Specifies that a valid log archive destination can be used for a subsequent archiving operation (automatic or manual). This is the default.

- `defer`

  Specifies that valid destination information and attributes are preserved, but the destination is excluded from archiving operations until re-enabled.

- `alternate`

  Specifies that a log archive destination is not enabled but will become enabled if communications to another destination fail.

The LOG_ARCHIVE_DEST_STATE_*n* parameters have no effect on the ENABLE state for the LOG_ARCHIVE_DEST or LOG_ARCHIVE_DUPLEX_DEST parameters.

The V$ARCHIVE_DEST dynamic performance view shows values in use for the current session. The DEST_ID column of that view corresponds to the archive destination suffix *n*.

See Also:

- *Oracle Database Backup and Recovery User's Guide* and *Oracle Database Administrator's Guide* for more information and examples
- "V$ARCHIVE_DEST" on page 6-8

LOG_ARCHIVE_DUPLEX_DEST

Property	Description
Parameter type	String
Syntax	`LOG_ARCHIVE_DUPLEX_DEST =` *`filespec`*
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Range of values	Either a null string or any valid path or device name, except raw partitions
Basic	No

> **Note:** If you are using Oracle Enterprise Edition, this parameter is deprecated in favor of the `LOG_ARCHIVE_DEST_`*`n`* parameters. If Oracle Enterprise Edition is not installed or it is installed but you have not specified any `LOG_ARCHIVE_DEST_`*`n`* parameters, this parameter is valid.

`LOG_ARCHIVE_DUPLEX_DEST` is similar to the initialization parameter `LOG_ARCHIVE_DEST`. This parameter specifies a second archive destination: the **duplex** archive destination. This duplex archive destination can be either a must-succeed or a best-effort archive destination, depending on how many archive destinations must succeed (as specified in the `LOG_ARCHIVE_MIN_SUCCEED_DEST` parameter).

The default setting of a null string ("") or (' ') indicates that a duplex archive destination does not exist.

See Also:

- "LOG_ARCHIVE_DEST_n" on page 1-81
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-89
- "V$ARCHIVE_DEST" on page 6-8

LOG_ARCHIVE_FORMAT

Property	Description
Parameter type	String
Syntax	`LOG_ARCHIVE_FORMAT =` *`filename`*
Default value	Operating system-dependent
Modifiable	No
Range of values	Any string that resolves to a valid filename
Basic	No
Real Application Clusters	Multiple instances can have different values, but identical values are recommended.

`LOG_ARCHIVE_FORMAT` is applicable only if you are using the redo log in `ARCHIVELOG` mode. Use a text string and variables to specify the default filename format when archiving redo log files. The string generated from this format is appended to the string specified in the LOG_ARCHIVE_DEST parameter.

The following variables can be used in the format:

%s log sequence number

%S log sequence number, zero filled

%tthread number

%Tthread number, zero filled

%a activation ID

%d database ID

%r resetlogs ID that ensures unique names are constructed for the archived log files across multiple incarnations of the database

Using uppercase letters for the variables (for example, %S) causes the value to be fixed length and padded to the left with zeros. An example of specifying the archive redo log filename format follows:

```
LOG_ARCHIVE_FORMAT = 'log%t_%s_%r.arc'
```

Neither LOG_ARCHIVE_DEST nor LOG_ARCHIVE_FORMAT have to be complete file or directory specifiers themselves; they only need to form a valid file path after the variables are substituted into LOG_ARCHIVE_FORMAT and the two parameters are concatenated together.

See Also:

- *Oracle Database Backup and Recovery Basics*, *Oracle Data Guard Concepts and Administration*, and *Oracle Real Application Clusters Administration and Deployment Guide* for more information on setting this parameter
- Your operating system- specific Oracle documentation for the default value and range of values for LOG_ARCHIVE_FORMAT

LOG_ARCHIVE_LOCAL_FIRST

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SYSTEM`
Range of values	`true | false`
Basic	No

Note: The STANDBY_ARCHIVE_DEST parameter is deprecated. It is retained for backward compatibility only.

LOG_ARCHIVE_LOCAL_FIRST specifies when the archiver processes (ARC*n*) transmit redo data to remote standby database destinations.

Values:

- `true`

  Directs the ARC*n* process to transmit redo data after the online redo log file has been completely and successfully archived to at least one local destination. This is the default value.

  Because the online redo log files are archived locally first, the LGWR process reuses the online redo log files much earlier than would be possible if the ARC*n* processes archived to the standby database concurrently with the local destination. This behavior is useful when archiving to remote destinations that use a slow network connection, such as a long-distance wide area network (WAN).

- `false`

  Directs the ARC*n* process to transmit redo data at the same time the online redo log file is archived to the local destinations. This results in redo data being promptly dispatched to the remote standby database destination.

  Setting `LOG_ARCHIVE_LOCAL_FIRST` to `false` is most useful for faster network connections, such as high-speed local area networks (LAN).

If `LOG_ARCHIVE_LOCAL_FIRST` is set to `true`, then it is not used on a physical standby database and any database for which the following attributes have been specified in the `LOG_ARCHIVE_DEST_`*n* initialization parameter:

- `MANDATORY`
- `LOCAL`
- `DEPENDENCY` (or any database destination that is the target of a `DEPENDENCY` attribute)

If `LOG_ARCHIVE_LOCAL_FIRST` is set to `true`, then it is ignored during certain operations, such as during a switchover, which requires synchronized archival operations. If the destination was explicitly configured to use the log writer process (by specifying the `LGWR` attribute in the `LOG_ARCHIVE_DEST_`*n* initialization parameter), but for some reason the log writer process becomes unable to archive to the destination, then Data Guard will revert to using the ARC*n* process to complete archival operations using the default behavior, even if `LOG_ARCHIVE_LOCAL_FIRST` is set to `false`.

For example, if a standby database problem or a network problem causes the LGWR process to fail, then the ARC*n* process will complete the archival. Data Guard minimizes the effect on the primary database as much as possible by archiving to the local destination first to ensure the online redo log files are available to the LGWR process as quickly as possible.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about the `LOG_ARCHIVE_LOCAL_FIRST` initialization parameter and its effect on ARC*n* behavior

LOG_ARCHIVE_MAX_PROCESSES

Property	Description
Parameter type	Integer
Default value	`4`
Modifiable	`ALTER SYSTEM`
Range of values	`1` to `30`

Property	Description
Basic	No

LOG_ARCHIVE_MAX_PROCESSES specifies the number of archiver background processes (ARC0 through ARC*n*) Oracle initially invokes.

- If the LOG_ARCHIVE_START initialization parameter has the value true, then this value is evaluated at instance startup. Note that the LOG_ARCHIVE_START initialization parameter was deprecated as of Oracle Database 10g Release 1 (10.1).
- Otherwise, this parameter is evaluated when the archiver process is first invoked by SQL*Plus or SQL syntax.

The actual number of archiver processes in use may vary subsequently based on archive workload.

See Also: *Oracle Database Concepts*

LOG_ARCHIVE_MIN_SUCCEED_DEST

Property	Description
Parameter type	Integer
Default value	1
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	1 to 10 if you are using LOG_ARCHIVE_DEST_*n* 1 or 2 if you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST
Basic	No

LOG_ARCHIVE_MIN_SUCCEED_DEST defines the minimum number of destinations that must succeed in order for the online logfile to be available for reuse.

- If you are using the LOG_ARCHIVE_DEST_*n* parameters and automatic archiving is enabled, the value of this parameter cannot exceed either of the following:

  - The total number of destinations

  - The number of enabled, valid destinations specified as MANDATORY plus the number of enabled, valid non-standby destinations specified as OPTIONAL
- If you are using LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST and automatic archiving is enabled, a value of 1 specifies that the destination specified in LOG_ARCHIVE_DEST must succeed. A value of 2 specifies that the destinations specified in both parameters must succeed.

If the value of this parameter is less than the number of enabled, valid MANDATORY destinations, this parameter is ignored in favor of the MANDATORY destination count. If the value is more than the number of enabled, valid MANDATORY destinations, some of the enabled, valid OPTIONAL non-standby destinations are treated as MANDATORY.

You can switch dynamically from using the older parameters to the LOG_ARCHIVE_DEST_*n* parameter using ALTER SYSTEM, as follows:

1. Set LOG_ARCHIVE_MIN_SUCCEED_DEST to 1.

2. Set the value of LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST to the null string.
3. Set the desired number of destinations for the LOG_ARCHIVE_DEST_*n* parameters.
4. Reset LOG_ARCHIVE_MIN_SUCCEED_DEST to the desired value.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- "LOG_ARCHIVE_DEST_n" on page 1-81, "LOG_ARCHIVE_DUPLEX_DEST" on page 1-86, and "V$ARCHIVE_DEST" on page 6-8 for information on related parameters

LOG_ARCHIVE_TRACE

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192
Basic	No
Real Application Clusters	Multiple instances can have different values.

LOG_ARCHIVE_TRACE controls output generated by the archivelog process.

This process can be initiated by any of the following:

- An ARC*n* background process (designated as ARC*n* in the output logs)
- An explicit session-invoked foreground process (designated as ARCH in the output logs)

The valid values have the following meanings:

- 0: Disable archivelog tracing (this is the default)
- 1: Track archival of redo log file
- 2: Track archival status of each archivelog destination
- 4: Track archival operational phase
- 8: Track archivelog destination activity
- 16: Track detailed archivelog destination activity
- 32: Track archivelog destination parameter modifications
- 64: Track ARC*n* process state activity
- 128: Track FAL (fetch archived log) server related activities
- 256: Track RFS Logical Client
- 512: Track LGWR redo shipping network activity

- 1024: Track RFS Physical Client
- 2048: Track RFS/ARCn Ping Heartbeat
- 4096: Track Real Time Apply
- 8192: Track Redo Apply (Media Recovery or Physical Standby)

You can combine tracing levels by adding together the values of the desired tracing levels. For example, a setting of 3 will generate level 1 and level 2 trace output. You can set different values for the primary and standby database.

When this parameter is set to the default value of 0, Oracle will still generate appropriate alert and trace entries in response to error conditions. If you change the value of this parameter dynamically in an `ALTER SYSTEM` statement, the changes will take effect at the start of the next archivelog operation.

See Also: *Oracle Database Administrator's Guide*

LOG_BUFFER

Property	Description
Parameter type	Integer
Default value	512 KB or 128 KB * `CPU_COUNT`, whichever is greater
Modifiable	No
Range of values	Operating system-dependent
Basic	No

`LOG_BUFFER` specifies the amount of memory (in bytes) that Oracle uses when buffering redo entries to a redo log file. Redo log entries contain a record of the changes that have been made to the database block buffers. The LGWR process writes redo log entries from the log buffer to a redo log file.

In general, larger values for `LOG_BUFFER` reduce redo log file I/O, particularly if transactions are long or numerous. In a busy system, a value 65536 or higher is reasonable.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

LOG_CHECKPOINT_INTERVAL

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SYSTEM`
Range of values	0 to 2^{31} - 1

Property	Description
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LOG_CHECKPOINT_INTERVAL` specifies the frequency of checkpoints in terms of the number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log. This number refers to physical operating system blocks, not database blocks.

Regardless of this value, a checkpoint always occurs when switching from one online redo log file to another. Therefore, if the value exceeds the actual redo log file size, checkpoints occur only when switching logs. Checkpoint frequency is one of the factors that influence the time required for the database to recover from an unexpected failure.

Notes:

- Specifying a value of `0` (zero) for `LOG_CHECKPOINT_INTERVAL` has the same effect as setting the parameter to infinity and causes the parameter to be ignored. Only nonzero values of this parameter are considered meaningful.
- Recovery I/O can also be limited by setting the `LOG_CHECKPOINT_TIMEOUT` parameter or by the size specified for the smallest redo log. For information on which mechanism is controlling checkpointing behavior, query the `V$INSTANCE_RECOVERY` view.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "LOG_CHECKPOINT_TIMEOUT" on page 1-92, and "V$INSTANCE_RECOVERY" on page 7-3

LOG_CHECKPOINT_TIMEOUT

Property	Description
Parameter type	Integer
Default value	`1800`
Modifiable	`ALTER SYSTEM`
Range of values	`0` to 2^{31} - 1
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LOG_CHECKPOINT_TIMEOUT` specifies (in seconds) the amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log (sometimes called the **tail of the log**) occurred. This parameter also signifies that no buffer will remain dirty (in the cache) for more than *integer* seconds.

Specifying a value of 0 for the timeout disables time-based checkpoints. Hence, setting the value to 0 is not recommended unless FAST_START_MTTR_TARGET is set.

Notes:

- A checkpoint scheduled to occur because of this parameter is delayed until the completion of the previous checkpoint if the previous checkpoint has not yet completed.
- Recovery I/O can also be limited by setting the LOG_CHECKPOINT_INTERVAL parameter or by the size specified for the smallest redo log. For information on which mechanism is controlling checkpointing behavior, query the V$INSTANCE_RECOVERY view.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "LOG_CHECKPOINT_TIMEOUT" on page 1-92, and "V$INSTANCE_RECOVERY" on page 7-3

LOG_CHECKPOINTS_TO_ALERT

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SYSTEM
Range of values	true | false
Basic	No

LOG_CHECKPOINTS_TO_ALERT lets you log your checkpoints to the alert file. Doing so is useful for determining whether checkpoints are occurring at the desired frequency.

LOG_FILE_NAME_CONVERT

Property	Description
Parameter type	String

Property	Description

Syntax	`LOG_FILE_NAME_CONVERT = 'string1' , 'string2' , 'string3' , 'string4' , ...`
	Where:
	- *string1* is the pattern of the primary database filename
	- *string2* is the pattern of the standby database filename
	- *string3* is the pattern of the primary database filename
	- *string4* is the pattern of the standby database filename
	You can use as many pairs of primary and standby replacement strings as required. You can use single or double quotation marks.
	The following are example settings that are acceptable:
	`LOG_FILE_NAME_CONVERT = '/dbs/t1/','/dbs/t1/s_','dbs/t2/ ','dbs/t2/s_'`

Default value	There is no default value.
Modifiable	`ALTER SESSION`
Basic	No

`LOG_FILE_NAME_CONVERT` converts the filename of a new log file on the primary database to the filename of a log file on the standby database. If you add a log file to the primary database, you must add a corresponding file to the standby database.

If you specify an odd number of strings (the last string has no corresponding replacement string), an error is signalled during startup. If the filename being converted matches more than one pattern in the pattern/replace string list, the first matched pattern takes effect. There is no limit on the number of pairs that you can specify in this parameter (other than the hard limit of the maximum length of multivalue parameters).

When the standby database is updated, this parameter converts the log file name on the primary database to the log file name on the standby database. The file must exist on the standby database and must be writable or the recovery process will halt with an error.

The first string is the pattern found in the log file names on the primary database. The second string is the pattern found in the log file names on the standby database.

You should also use `LOG_FILE_NAME_CONVERT` to rename the logfiles in the clone control file when setting up the clone database during tablespace point-in-time recovery.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* and *Oracle Data Guard Concepts and Administration*

MAX_COMMIT_PROPAGATION_DELAY

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Range of values	0 to 90000

Property	Description
Basic	No
Real Application Clusters	Multiple instances must have the same value.

Note: The `MAX_COMMIT_PROPAGATION_DELAY` parameter is deprecated. It is retained for backward compatibility only.

`MAX_COMMIT_PROPAGATION_DELAY` is a Real Application Clusters parameter. This initialization parameter should not be changed except under a limited set of circumstances specific to the cluster database.

MAX_DISPATCHERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Range of values	If `MAX_DISPATCHERS` is specified, then it should be greater than or equal to the number of dispatchers specified by the `DISPATCHERS` parameter and less than the number of processes specified by the `PROCESSES` parameter.
Basic	No

`MAX_DISPATCHERS` specifies the maximum number of dispatcher processes allowed to be running simultaneously. It can be overridden by the `DISPATCHERS` parameter and is maintained for backward compatibility with older releases.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value and range of values

MAX_DUMP_FILE_SIZE

Property	Description
Parameter type	String
Syntax	`MAX_DUMP_FILE_SIZE = {` *integer* `[K	M	G]	UNLIMITED }`
Default value	`UNLIMITED`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	0 to unlimited, or `UNLIMITED`
Basic	No

MAX_DUMP_FILE_SIZE specifies the maximum size of trace files (excluding the alert file). Change this limit if you are concerned that trace files may use too much space.

- A numerical value for MAX_DUMP_FILE_SIZE specifies the maximum size in operating system blocks.
- A number followed by a K or M suffix specifies the file size in kilobytes or megabytes.
- The special value string UNLIMITED means that there is no upper limit on trace file size. Thus, dump files can be as large as the operating system permits.

> **See Also:** *Oracle Database Administrator's Guide* and *Oracle Database Performance Tuning Guide* for more information on setting this parameter

MAX_ENABLED_ROLES

Property	Description
Parameter type	Integer
Default value	30
Modifiable	No
Range of values	0 to 148
Basic	No

> **Note:** The MAX_ENABLED_ROLES parameter is deprecated. It is retained for backward compatibility only.

MAX_ENABLED_ROLES specifies the maximum number of database roles that users can enable, including roles contained within other roles. The actual number of roles that users can enable is 2 plus the value of MAX_ENABLED_ROLES, because each user has two additional roles, PUBLIC and the user's own role.

For example, if MAX_ENABLED_ROLES is set to 5, then user HR can have seven roles enabled (the five enabled by MAX_ENABLED_ROLES, plus PUBLIC and HR).

MAX_SHARED_SERVERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If MAX_SHARED_SERVERS is specified, then it should be greater than or equal to SHARED_SERVERS and less than PROCESSES.
Basic	No

MAX_SHARED_SERVERS specifies the maximum number of shared server processes allowed to be running simultaneously. Setting this parameter enables you to reserve process slots for other processes, such as dedicated servers.

When you want to reduce the range of shared servers, you can reduce MAX_SHARED_SERVERS before reducing SHARED_SERVERS. If MAX_SHARED_SERVERS is lower than SHARED_SERVERS, then the number of shared servers will not vary but will remain at the constant level specified by SHARED_SERVERS. If MAX_SHARED_SERVERS is not specified, then a shared server process may be spawned as long as the number of free process slots is greater than 1 / 8 the maximum number of processes, or 2 if PROCESSES is less than 24.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Database Concepts* for information on processes
- Your operating system-specific Oracle documentation for the default value and range of values

MEMORY_MAX_TARGET

Property	Description
Parameter type	Big integer
Syntax	MEMORY_MAX_TARGET = *integer* [K | M | G]
Default value	0
Modifiable	No
Range of values	0 to the physical memory size available to the Oracle Database
Basic	No

MEMORY_MAX_TARGET specifies the maximum value to which a DBA can set the MEMORY_TARGET initialization parameter. See the description of MEMORY_TARGET for more information about how the settings of MEMORY_MAX_TARGET and MEMORY_TARGET affect each other.

See Also: *Oracle Database Administrator's Guide* for more information about managing memory

MEMORY_TARGET

Property	Description
Parameter type	Big integer
Syntax	MEMORY_TARGET = *integer* [K | M | G]
Default value	0
Modifiable	ALTER SYSTEM
Range of values	152 MB to MEMORY_MAX_TARGET
Basic	No

MEMORY_TARGET specifies the Oracle system-wide usable memory. The database tunes memory to the MEMORY_TARGET value, reducing or enlarging the SGA and PGA as needed.

In a text-based initialization parameter file, if you omit MEMORY_MAX_TARGET and include a value for MEMORY_TARGET, then the database automatically sets MEMORY_MAX_TARGET to the value of MEMORY_TARGET. If you omit the line for MEMORY_TARGET and include a value for MEMORY_MAX_TARGET, the MEMORY_TARGET parameter defaults to zero. After startup, you can then dynamically change MEMORY_TARGET to a nonzero value, provided that it does not exceed the value of MEMORY_MAX_TARGET.

See Also: *Oracle Database Administrator's Guide* for more information about managing memory

NLS_CALENDAR

Property	Description
Parameter type	String
Syntax	NLS_CALENDAR = "*calendar_system*"
Default value	None
Modifiable	ALTER SESSION
Range of values	Any valid calendar format name
Basic	No

NLS_CALENDAR specifies which calendar system Oracle uses. It can have one of the following values:

- Arabic Hijrah
- English Hijrah
- Gregorian
- Japanese Imperial
- Persian
- ROC Official (Republic of China)
- Thai Buddha

For example, suppose NLS_CALENDAR is set to "Japanese Imperial", the date format is "E YY-MM-DD". ("E" is the date format element for the abbreviated era name.) If the date is May 15, 1997, then the SYSDATE is displayed as follows:

```
SELECT SYSDATE FROM DUAL;
SYSDATE
--------
H 09-05-15
```

See Also: *Oracle Database Globalization Support Guide* for a listing of available calendar systems

NLS_COMP

Property	Description
Parameter type	String
Syntax	`NLS_COMP = { BINARY | LINGUISTIC | ANSI }`
Default value	`BINARY`
Modifiable	`ALTER SESSION`
Basic	No

`NLS_COMP` specifies the collation behavior of the database session.

Values:

- `BINARY`

  Normally, comparisons in the `WHERE` clause and in PL/SQL blocks is binary unless you specify the `NLSSORT` function.

- `LINGUISTIC`

  Comparisons for all SQL operations in the `WHERE` clause and in PL/SQL blocks should use the linguistic sort specified in the `NLS_SORT` parameter. To improve the performance, you can also define a linguistic index on the column for which you want linguistic comparisons.

- `ANSI`

  A setting of `ANSI` is for backwards compatibility; in general, you should set `NLS_COMP` to `LINGUISTIC`

Note: Unless you explicitly set the value for `NLS_COMP` in your initialization parameter file, a default value of NULL is shown in the following views: `V$PARAMETER`, `V$SYSTEM_PARAMETER`, `V$PARAMETER2`, `V$SYSTEM_PARAMETER2`, and `NLS_INSTANCE_PARAMETERS`. However, the actual default value, and behavior, is `BINARY`. Note that you cannot change the default to NULL, because NULL is not among the valid values.

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_CURRENCY

Property	Description
Parameter type	String
Syntax	`NLS_CURRENCY =` *`currency_symbol`*
Default value	Derived from `NLS_TERRITORY`
Modifiable	`ALTER SESSION`
Range of values	Any valid character string, with a maximum of 10 bytes (not including null)

Property	Description
Basic	No

`NLS_CURRENCY` specifies the string to use as the local currency symbol for the L number format element. The default value of this parameter is determined by `NLS_TERRITORY`.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Language Reference* for information on number format elements

NLS_DATE_FORMAT

Property	Description
Parameter type	String
Syntax	`NLS_DATE_FORMAT = "`*`format`*`"`
Default value	Derived from `NLS_TERRITORY`
Modifiable	`ALTER SESSION`
Range of values	Any valid date format mask but not exceeding a fixed length
Basic	No

`NLS_DATE_FORMAT` specifies the default date format to use with the `TO_CHAR` and `TO_DATE` functions. The default value of this parameter is determined by `NLS_TERRITORY`.

The value of this parameter can be any valid date format mask, and the value must be surrounded by double quotation marks. For example:

```
NLS_DATE_FORMAT = "MM/DD/YYYY"
```

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DATE_LANGUAGE

Property	Description
Parameter type	String
Syntax	`NLS_DATE_LANGUAGE =` *`language`*
Default value	Derived from `NLS_LANGUAGE`
Modifiable	`ALTER SESSION`

Property	Description
Range of values	Any valid NLS_LANGUAGE value

NLS_DATE_LANGUAGE specifies the language to use for the spelling of day and month names and date abbreviations (a.m., p.m., AD, BC) returned by the TO_DATE and TO_CHAR functions.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Language Reference* for information on the TO_DATE and TO_CHAR functions.
- *Oracle Database Heterogeneous Connectivity Administrator's Guide* for information on setting this parameter in heterogeneous systems

NLS_DUAL_CURRENCY

Property	Description
Parameter type	String
Syntax	NLS_DUAL_CURRENCY = *currency_symbol*
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid format name up to 10 characters

NLS_DUAL_CURRENCY specifies the dual currency symbol (such as "Euro") for the territory. The default is the dual currency symbol defined in the territory of your current language environment.

See Also: *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_ISO_CURRENCY

Property	Description
Parameter type	String
Syntax	NLS_ISO_CURRENCY = *territory*
Default value	Derived from NLS_TERRITORY
Modifiable	ALTER SESSION
Range of values	Any valid NLS_TERRITORY value

NLS_ISO_CURRENCY specifies the string to use as the international currency symbol for the C number format element.

Local currency symbols can be ambiguous. For example, a dollar sign ($) can refer to U.S. dollars or Australian dollars. ISO Specification 4217 1987-07-15 defines unique "international" currency symbols for the currencies of specific territories or countries.

See Also:

- *Oracle Database Globalization Support Guide* for more information on setting this parameter
- *Oracle Database SQL Language Reference* for information on number format elements

NLS_LANGUAGE

Property	Description
Parameter type	String
Syntax	`NLS_LANGUAGE = `*`language`*
Default value	Operating system-dependent, derived from the `NLS_LANG` environment variable
Modifiable	`ALTER SESSION`
Range of values	Any valid language name
Basic	Yes

`NLS_LANGUAGE` specifies the default language of the database. This language is used for messages, day and month names, symbols for AD, BC, a.m., and p.m., and the default sorting mechanism. This parameter also determines the default values of the parameters `NLS_DATE_LANGUAGE` and `NLS_SORT`.

See Also:

- *Oracle Database Globalization Support Guide* for a complete list of languages and additional information on this parameter
- Your operating system-specific Oracle documentation and the release notes for your country

NLS_LENGTH_SEMANTICS

Property	Description
Parameter type	String
Syntax	`NLS_LENGTH_SEMANTICS = `*`string`* Example: `NLS_LENGTH_SEMANTICS = 'CHAR'`
Default value	`BYTE`
Modifiable	`ALTER SESSION`
Range of values	`BYTE` | `CHAR`

`NLS_LENGTH_SEMANTICS` enables you to create `CHAR` and `VARCHAR2` columns using either byte or character length semantics. Existing columns are not affected.

`NCHAR`, `NVARCHAR2`, `CLOB`, and `NCLOB` columns are always character-based. You may be required to use byte semantics in order to maintain compatibility with existing applications.

`NLS_LENGTH_SEMANTICS` does not apply to tables in the `SYS` schema.

NLS_NCHAR_CONV_EXCP

Property	Description
Parameter type	String
Syntax	`NLS_NCHAR_CONV_EXCP = { TRUE | FALSE }`
Default value	`FALSE`
Modifiable	`ALTER SESSION`

`NLS_NCHAR_CONV_EXCP` determines whether an error is reported when there is data loss during an implicit or explicit character type conversion between NCHAR/NVARCHAR and CHAR/VARCHAR2. The default value results in no error being reported.

> **See Also:** *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_NUMERIC_CHARACTERS

Property	Description
Parameter type	String

Syntax	`NLS_NUMERIC_CHARACTERS =`
	`"`*`decimal_character group_separator`*`"`

Default value	Derived from `NLS_TERRITORY`
Modifiable	`ALTER SESSION`

`NLS_NUMERIC_CHARACTERS` specifies the characters to use as the group separator and decimal character. It overrides those characters defined implicitly by `NLS_TERRITORY`. The group separator separates integer groups (that is, thousands, millions, billions, and so on). The decimal separates the integer portion of a number from the decimal portion.

You can specify any character as the decimal or group separator. The two characters specified must be single-byte and must be different from each other. The characters cannot be any numeric character or any of the following characters: plus (+), minus sign (-), less than sign (<), greater than sign (>). Either character can be a space.

For example, if you wish to specify a comma as the decimal character and a space as the group separator, you would set this parameter as follows:

```
NLS_NUMERIC_CHARACTERS = ", "
```

> **See Also:** *Oracle Database Globalization Support Guide* for more information on setting this parameter

NLS_SORT

Property	Description
Parameter type	String

Property	Description
Syntax	`NLS_SORT = { BINARY | *linguistic_definition* }`
Default value	Derived from `NLS_LANGUAGE`
Modifiable	`ALTER SESSION`
Range of values	`BINARY` or any valid linguistic definition name

`NLS_SORT` specifies the collating sequence for `ORDER BY` queries.

- If the value is `BINARY`, then the collating sequence for `ORDER BY` queries is based on the numeric value of characters (a binary sort that requires less system overhead).
- If the value is a named linguistic sort, sorting is based on the order of the defined linguistic sort. Most (but not all) languages supported by the `NLS_LANGUAGE` parameter also support a linguistic sort with the same name.

> **Note:** Setting `NLS_SORT` to anything other than `BINARY` causes a sort to use a full table scan, regardless of the path chosen by the optimizer. `BINARY` is the exception because indexes are built according to a binary order of keys. Thus the optimizer can use an index to satisfy the `ORDER BY` clause when `NLS_SORT` is set to `BINARY`. If `NLS_SORT` is set to any linguistic sort, the optimizer must include a full table scan and a full sort in the execution plan.

You must use the `NLS_SORT` operator with comparison operations if you want the linguistic sort behavior.

See Also:

- *Oracle Database Globalization Support Guide* for a current listing of values you can specify for this parameter
- *Oracle Database Concepts* for more information on this parameter

NLS_TERRITORY

Property	Description
Parameter type	String
Syntax	`NLS_TERRITORY = *territory*`
Default value	Operating system-dependent
Modifiable	`ALTER SESSION`
Range of values	Any valid territory name
Basic	Yes

`NLS_TERRITORY` specifies the name of the territory whose conventions are to be followed for day and week numbering.

This parameter also establishes the default date format, the default decimal character and group separator, and the default ISO and local currency symbols.

For information on these settings, see "NLS_DATE_FORMAT" on page 1-100, "NLS_NUMERIC_CHARACTERS" on page 1-103, "NLS_CURRENCY" on page 1-99, and "NLS_ISO_CURRENCY" on page 1-101.

See Also:

- *Oracle Database Globalization Support Guide* for a complete list of territories
- Your operating system-specific Oracle documentation for the territory-dependent default values for these parameters

NLS_TIMESTAMP_FORMAT

Property	Description
Parameter type	String
Syntax	`NLS_TIMESTAMP_FORMAT = "`*`format`*`"`
Default value	Derived from `NLS_TERRITORY`
Modifiable	`ALTER SESSION`
Range of values	Any valid datetime format mask

`NLS_TIMESTAMP_FORMAT` defines the default timestamp format to use with the `TO_CHAR` and `TO_TIMESTAMP` functions.

The value must be surrounded by quotation marks as follows:

```
NLS_TIMESTAMP_FORMAT = 'YYYY-MM-DD HH:MI:SS.FF'
```

You can specify the value of `NLS_TIMESTAMP_FORMAT` by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of `NLS_TIMESTAMP_FORMAT` by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the `ALTER SESSION SET` statement.

NLS_TIMESTAMP_TZ_FORMAT

Property	Description
Parameter type	String
Syntax	`NLS_TIMESTAMP_TZ_FORMAT = "`*`format`*`"`
Default value	Derived from `NLS_TERRITORY`
Modifiable	`ALTER SESSION`
Range of values	Any valid datetime format mask

`NLS_TIMESTAMP_TZ_FORMAT` defines the default timestamp with time zone format to use with the `TO_CHAR` and `TO_TIMESTAMP_TZ`functions.

The value must be surrounded by quotation marks as follows:

```
NLS_TIMESTAMP_TZ_FORMAT = 'YYYY-MM-DD HH:MI:SS.FF TZH:TZM'
```

You can specify the value of NLS_TIMESTAMP_TZ_FORMAT by setting it in the initialization parameter file. You can specify its value for a client as a client environment variable.

You can also alter the value of NLS_TIMESTAMP_TZ_FORMAT by changing its value in the initialization parameter and then restarting the instance. To alter the value during a session use the ALTER SESSION SET statement.

O7_DICTIONARY_ACCESSIBILITY

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true | false

O7_DICTIONARY_ACCESSIBILITY controls restrictions on SYSTEM privileges. If the parameter is set to true, access to objects in the SYS schema is allowed (Oracle7 behavior). The default setting of false ensures that system privileges that allow access to objects in "any schema" do not allow access to objects in the SYS schema.

For example, if O7_DICTIONARY_ACCESSIBILITY is set to false, then the SELECT ANY TABLE privilege allows access to views or tables in any schema except the SYS schema (data dictionary tables cannot be accessed). The system privilege EXECUTE ANY PROCEDURE allows access on the procedures in any schema except the SYS schema.

If this parameter is set to false and you need to access objects in the SYS schema, then you must be granted explicit object privileges. The following roles, which can be granted to the database administrator, also allow access to dictionary objects:

- SELECT_CATALOG_ROLE
- EXECUTE_CATALOG_ROLE
- DELETE_CATALOG_ROLE

See Also:

- *Oracle Database Upgrade Guide* for more information on this parameter and the roles mentioned here
- *Oracle Database SQL Language Reference* for information on granting roles

OBJECT_CACHE_MAX_SIZE_PERCENT

Property	Description
Parameter type	Integer
Default value	10
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	0 to operating system-dependent maximum

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server. `OBJECT_CACHE_MAX_SIZE_PERCENT` specifies the percentage of the optimal cache size that the session object cache can grow past the optimal size. The maximum size is equal to the optimal size plus the product of this percentage and the optimal size. When the cache size exceeds this maximum size, the system will attempt to shrink the cache to the optimal size.

See Also:

- "OBJECT_CACHE_OPTIMAL_SIZE" on page 1-107 for a description of the object cache
- *Oracle Database Concepts*, *Pro*C/C++ Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OBJECT_CACHE_OPTIMAL_SIZE

Property	Description
Parameter type	Integer
Default value	`102400` (100K)
Modifiable	`ALTER SESSION`, `ALTER SYSTEM ... DEFERRED`
Range of values	10 KB to operating system-dependent maximum

The **object cache** is a memory block on the client that allows applications to store entire objects and to navigate among them without round trips to the server. `OBJECT_CACHE_OPTIMAL_SIZE` specifies (in bytes) the size to which the session object cache is reduced when the size of the cache exceeds the maximum size.

See Also: *Oracle Database Concepts*, *Pro*C/C++ Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for information on precompiler use of the object cache

OLAP_PAGE_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	`OLAP_PAGE_POOL_SIZE =` *`integer`* `[K | M | G]`
Default value	0
Modifiable	`ALTER SESSION`, `ALTER SYSTEM ... DEFERRED`
Range of values	0 to 2 GB

`OLAP_PAGE_POOL_SIZE` specifies (in bytes) the size of the OLAP page pool.

See Also: *Oracle OLAP User's Guide* for more information about the OLAP page pool

OPEN_CURSORS

Property	Description
Parameter type	Integer
Default value	`50`
Modifiable	`ALTER SYSTEM`
Range of values	0 to 65535
Basic	Yes

`OPEN_CURSORS` specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. You can use this parameter to prevent a session from opening an excessive number of cursors.

It is important to set the value of `OPEN_CURSORS` high enough to prevent your application from running out of open cursors. The number will vary from one application to another. Assuming that a session does not open the number of cursors specified by `OPEN_CURSORS`, there is no added overhead to setting this value higher than actually needed.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the range of values

OPEN_LINKS

Property	Description
Parameter type	Integer
Default value	`4`
Modifiable	No
Range of values	0 to 255

`OPEN_LINKS` specifies the maximum number of concurrent open connections to remote databases in one session. These connections include database links, as well as external procedures and cartridges, each of which uses a separate process.

Oracle counts one open link for the following:

- For each user that references a public or private database link
- For each external procedure or cartridge connection when it is executed for the first time

Both types of connections close when the session ends. You can also close a database link connection explicitly by issuing an `ALTER SESSION CLOSE DATABASE LINK` statement.

You should set this parameter to allow for the external procedure and cartridge connections expected during the session plus the number of databases referred to in typical distributed transactions (that is, a single SQL statement that references multiple databases), so that all the databases can be open to execute the statement. For example, if queries alternately access databases A, B, and C, and `OPEN_LINKS` is set to `2`, time will be lost waiting while one connection is broken and another made. Increase the value if many different databases are accessed over time.

This parameter refers only to connections used for distributed transactions. Direct connections to a remote database specified as an application connects are not counted.

If you set `OPEN_LINKS` to `0`, then no distributed transactions are allowed.

> **See Also:** "OPEN_LINKS_PER_INSTANCE" on page 1-109 for information on setting open connections globally for a database instance

OPEN_LINKS_PER_INSTANCE

Property	Description
Parameter type	Integer
Default value	`4`
Modifiable	No
Range of values	0 to 4294967295 (4 GB -1)
Real Application Clusters	Multiple instances can have different values.

`OPEN_LINKS_PER_INSTANCE` specifies the maximum number of migratable open connections globally for each database instance. XA transactions use migratable open connections so that the connections are cached after a transaction is committed. Another transaction can use the connection, provided the user who created the connection is the same as the user who owns the transaction.

`OPEN_LINKS_PER_INSTANCE` is different from `OPEN_LINKS`, which indicates the number of connections from a session. The `OPEN_LINKS` parameter is not applicable to XA applications.

> **See Also:** "OPEN_LINKS" on page 1-108

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true` | `false`
Basic	No

`OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES` enables or disables the automatic recognition of repeatable SQL statements, as well as the generation of SQL plan baselines for such statements.

OPTIMIZER_DYNAMIC_SAMPLING

Property	Description
Parameter type	Integer
Default value	If `OPTIMIZER_FEATURES_ENABLE` is set to `10.0.0` or higher, then `2` If `OPTIMIZER_FEATURES_ENABLE` is set to `9.2.0`, then `1` If `OPTIMIZER_FEATURES_ENABLE` is set to `9.0.1` or lower, then `0`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`0` to `10`

`OPTIMIZER_DYNAMIC_SAMPLING` controls the level of dynamic sampling performed by the optimizer.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on setting this parameter

OPTIMIZER_FEATURES_ENABLE

Property	Description
Parameter type	String
Syntax	`OPTIMIZER_FEATURES_ENABLE = { 8.0.0 | 8.0.3 | 8.0.4 | 8.0.5 | 8.0.6 | 8.0.7 | 8.1.0 | 8.1.3 | 8.1.4 | 8.1.5 | 8.1.6 | 8.1.7 | 9.0.0 | 9.0.1 | 9.2.0 | 10.1.0 | 10.1.0.3 | 10.1.0.4 | 10.1.0.5 | 10.1.0.6 | 10.2.0.1 | 10.2.0.2 | 10.2.0.3 | 10.2.0.4 | 11.1.0.6}`
Default value	`11.1.0.6`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`OPTIMIZER_FEATURES_ENABLE` acts as an umbrella parameter for enabling a series of optimizer features based on an Oracle release number.

For example, if you upgrade your database from release 10.1 to release 11.1, but you want to keep the release 10.1 optimizer behavior, you can do so by setting this parameter to `10.1.0`. At a later time, you can try the enhancements introduced in releases up to and including release 11.1 by setting the parameter to `11.1.0.6`.

Table 1–3 describes some of the optimizer features that are enabled when you set the `OPTIMIZER_FEATURES_ENABLE` parameter to a 9.0 release or a 9.2 release.

Table 1–4 describes some of the optimizer features that are enabled when you set the `OPTIMIZER_FEATURES_ENABLE` parameter to a 10.1 release or a 10.2 release.

Table 1–5 describes some of the optimizer features that are enabled when you set the `OPTIMIZER_FEATURES_ENABLE` parameter to a 11.1 release.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about the optimizer and for information about the features listed in the following tables

Table 1–3 Optimizer Features for Oracle9I Releases

Features	9.0.0	9.0.1	9.2.0
Index fast full scan	X	X	X
Consideration of bitmap access to paths for tables with only B-tree indexes	X	X	X
Complex view merging	X	X	X
Peeking into user-defined bind variables	X	X	X
Index joins	X	X	X

Table 1–4 Optimizer Features for Oracle Database 10g Releases

Features	10.1.0	10.1.0.3	10.1.0.4	10.1.0.5	10.2.0.1	10.2.0.2
Index fast full scan	X	X	X	X	X	X
Consideration of bitmap access to paths for tables with only B-tree indexes	X	X	X	X	X	X
Complex view merging	X	X	X	X	X	X
Peeking into user-defined bind variables	X	X	X	X	X	X
Index joins	X	X	X	X	X	X
Dynamic sampling	X	X	X	X	X	X
Query rewrite enables	X	X	X	X	X	X
Skip unusable indexes	X	X	X	X	X	X
Automatically compute index statistics as part of creation	X	X	X	X	X	X
Cost-based query transformations	X	X	X	X	X	X
Allow rewrites with multiple MVs and/or base tables					X	X

Table 1–5 Optimizer Features for Oracle Database 11g Releases

Features	11.1.0.6
Adaptive cursor sharing	X
Use extended statistics to estimate selectivity	X
Use native implementation for full outer joins	X
Partition pruning using join filtering	X
Group by placement optimization	X
Null aware antijoins	X

OPTIMIZER_INDEX_CACHING

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	0 to 100

`OPTIMIZER_INDEX_CACHING` lets you adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

The cost of executing an index using an IN-list iterator or of executing a nested loops join when an index is used to access the inner table depends on the caching of that index in the buffer cache. The amount of caching depends on factors that the optimizer cannot predict, such as the load on the system and the block access patterns of different users.

You can modify the optimizer's assumptions about index caching for nested loops joins and IN-list iterators by setting this parameter to a value between 0 and 100 to indicate the percentage of the index blocks the optimizer should assume are in the cache. Setting this parameter to a higher value makes nested loops joins and IN-list iterators look less expensive to the optimizer. As a result, it will be more likely to pick nested loops joins over hash or sort-merge joins and to pick indexes using IN-list iterators over other indexes or full table scans. The default for this parameter is 0, which results in default optimizer behavior.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter

OPTIMIZER_INDEX_COST_ADJ

Property	Description
Parameter type	Integer
Default value	`100`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	1 to 10000

`OPTIMIZER_INDEX_COST_ADJ` lets you tune optimizer behavior for access path selection to be more or less index friendly—that is, to make the optimizer more or less prone to selecting an index access path over a full table scan.

The default for this parameter is 100 percent, at which the optimizer evaluates index access paths at the regular cost. Any other value makes the optimizer evaluate the access path at that percentage of the regular cost. For example, a setting of 50 makes the index access path look half as expensive as normal.

Note: The adjustment does not apply to user-defined cost functions for domain indexes.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter, and on its relationship to "OPTIMIZER_INDEX_CACHING"

OPTIMIZER_MODE

Property	Description
Parameter type	String

Property	Description

Syntax	`OPTIMIZER_MODE =`
	`{ first_rows_[1 | 10 | 100 | 1000] | first_rows | all_rows }`

Default value	`all_rows`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`OPTIMIZER_MODE` establishes the default behavior for choosing an optimization approach for the instance.

Values:

- `first_rows_`*`n`*

  The optimizer uses a cost-based approach and optimizes with a goal of best response time to return the first *n* rows (where *n* = 1, 10, 100, 1000).

- `first_rows`

  The optimizer uses a mix of costs and heuristics to find a best plan for fast delivery of the first few rows.

- `all_rows`

  The optimizer uses a cost-based approach for all SQL statements in the session and optimizes with a goal of best throughput (minimum resource use to complete the entire statement).

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Concepts* and *Oracle Database Performance Tuning Guide* for more information about the optimizer

OPTIMIZER_SECURE_VIEW_MERGING

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SYSTEM`
Range of values	`true | false`
Basic	No
Real Application Clusters	Multiple instances can have different values

`OPTIMIZER_SECURE_VIEW_MERGING` enables or disables view merging globally for the database.

Values:

- `false`

  Oracle does not use view merging or predicate move-around.

- `true`

  Oracle assesses the query, considering all transformations, and chooses the method with the lowest cost.

To take advantage of query rewrite for a particular query, you must disable the `OPTIMIZER_SECURE_VIEW_MERGING` parameter.

OPTIMIZER_USE_INVISIBLE_INDEXES

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`OPTIMIZER_USE_INVISIBLE_INDEXES` enables or disables the use of invisible indexes.

Values:

- `true`

  Invisible indexes are treated as visible (normal) indexes.

- `false`

  Invisible indexes will not be considered by the optimizer but will still be maintained by DML operations.

OPTIMIZER_USE_PENDING_STATISTICS

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`OPTIMIZER_USE_PENDING_STATISTICS` specifies whether or not the optimizer uses pending statistics when compiling SQL statements.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on setting this parameter

OPTIMIZER_USE_SQL_PLAN_BASELINES

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`OPTIMIZER_USE_SQL_PLAN_BASELINES` enables or disables the use of SQL plan baselines stored in SQL Management Base. When enabled, the optimizer looks for a SQL plan baseline for the SQL statement being compiled. If one is found in SQL Management Base, then the optimizer will cost each of the baseline plans and pick one with the lowest cost.

OS_AUTHENT_PREFIX

Property	Description
Parameter type	String
Syntax	`OS_AUTHENT_PREFIX =` *`authentication_prefix`*
Default value	`OPS$`
Modifiable	No

`OS_AUTHENT_PREFIX` specifies a prefix that Oracle uses to authenticate users attempting to connect to the server. Oracle concatenates the value of this parameter to the beginning of the user's operating system account name and password. When a connection request is attempted, Oracle compares the prefixed username with Oracle usernames in the database.

The default value of this parameter is `OPS$` for backward compatibility with previous versions. However, you might prefer to set the prefix value to "" (a null string), thereby eliminating the addition of any prefix to operating system account names.

Note: The text of the `OS_AUTHENT_PREFIX` parameter is case sensitive on some operating systems.

See Also:

- *Oracle Database Advanced Security Administrator's Guide* for more information on setting this parameter
- Your operating system-specific Oracle documentation for the default value

OS_ROLES

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

`OS_ROLES` determines whether Oracle or the operating system identifies and manages the roles of each username.

Values:

- `TRUE`

  The operating system completely manages the role grants for all database usernames. When a user attempts to create a session, the username's security domain is initialized using the roles identified by the operating system. A user can subsequently enable as many roles identified by the operating system as specified by the parameter `MAX_ENABLED_ROLES`.

  Revocation by Oracle of roles granted by the operating system is ignored, as are any roles previously granted by Oracle.

- `FALSE`

  Oracle identifies and manages the roles.

See Also:

- *Oracle Database Administrator's Guide* and *Oracle Database Advanced Security Administrator's Guide* for more information on roles and on setting this parameter
- "REMOTE_OS_ROLES" on page 1-134

PARALLEL_ADAPTIVE_MULTI_USER

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SYSTEM`
Range of values	`true` | `false`

`PARALLEL_ADAPTIVE_MULTI_USER`, when set to `true`, enables an adaptive algorithm designed to improve performance in multiuser environments that use parallel execution. The algorithm automatically reduces the requested degree of parallelism based on the system load at query startup time. The effective degree of parallelism is based on the default degree of parallelism, or the degree from the table or hints, divided by a reduction factor.

The algorithm assumes that the system has been tuned for optimal performance in a single-user environment.

Tables and hints use the default degree of parallelism.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution and on the algorithm described here

PARALLEL_AUTOMATIC_TUNING

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

> **Note:** The `PARALLEL_AUTOMATIC_TUNING` parameter is deprecated. It is retained for backward compatibility only.

When `PARALLEL_AUTOMATIC_TUNING` is set to `true`, Oracle determines the default values for parameters that control parallel execution. In addition to setting this parameter, you must specify the `PARALLEL` clause for the target tables in the system. Oracle then tunes all subsequent parallel operations automatically.

If you used parallel execution in a previous release and are now enabling `PARALLEL_AUTOMATIC_TUNING`, then you should reduce the amount of memory allocated from the shared pool to account for the decreased demand on that pool. This memory will now be allocated from the large pool, and will be computed automatically if `LARGE_POOL_SIZE` is not specified.

As part of the automatic tuning, Oracle will enable the `PARALLEL_ADAPTIVE_MULTI_USER` parameter. You can override any of the system-provided defaults if desired.

PARALLEL_EXECUTION_MESSAGE_SIZE

Property	Description
Parameter type	Integer
Default value	Operating system-dependent
Modifiable	No
Range of values	2148 to 65535 (64 KB - 1)
Real Application Clusters	Multiple instances must have the same value.

`PARALLEL_EXECUTION_MESSAGE_SIZE` specifies the size of messages for parallel execution (formerly referred to as parallel query, PDML, Parallel Recovery, replication).

On most platforms, the default value is `2148`bytes if `PARALLEL_AUTOMATIC_TUNING` is set to `false`, and `4096` bytes if `PARALLEL_AUTOMATIC_TUNING` is set to `true`.

The default value is adequate for most applications. Larger values require a larger shared pool. Larger values result in better performance at the cost of higher memory use. For this reason, replication gets no benefit from increasing the size.

Note: When `PARALLEL_AUTOMATIC_TUNING` is set to `TRUE`, message buffers are allocated out of the large pool. In this case, the default is generally higher. Note that the `PARALLEL_AUTOMATIC_TUNING` parameter has been deprecated.

PARALLEL_INSTANCE_GROUP

Property	Description
Parameter type	String
Syntax	`PARALLEL_INSTANCE_GROUP =` *`group_name`*
Default value	A group consisting of all instances currently active
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	Any group name specified in the `INSTANCE_GROUPS` parameter of any active instance
Real Application Clusters	Multiple instances can have different values.

`PARALLEL_INSTANCE_GROUP` is a Real Application Clusters parameter that you can specify in parallel mode only. Used in conjunction with the `INSTANCE_GROUPS` parameter, it lets you restrict parallel query operations to a limited number of instances.

This parameter identifies the parallel instance group Oracle will use for spawning parallel execution processes. Parallel operations will spawn parallel execution processes only on instances that specify a matching group in their `INSTANCE_GROUPS` parameter.

If the value assigned to `PARALLEL_INSTANCE_GROUP` is the name of a group that does not exist, then the operation runs serially. No parallelism is used.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for more information on parallel query execution in a Real Application Clusters environment

PARALLEL_IO_CAP_ENABLED

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`
Basic	No
Real Application Clusters	Multiple instances can have different values.

PARALLEL_IO_CAP_ENABLED specifies whether or not Oracle caps the default degree of parallelism to no greater than what the I/O system can support. This new value is calculated based on the results of the resource manager's I/O calibration package.

If PARALLEL_IO_CAP_ENABLED is set to true and the database I/O capacity has been calibrated, then Oracle will reduce the default degree of parallelism if the I/O capacity cannot support the available number of CPUs. To calibrate the I/O capacity, use the DBMS_RESOURCE_MANAGER.CALIBRATE_IO procedure to measure the system's I/O capacity. Oracle then computes the degree of parallelism so as not to exceed the system's I/O capacity.

See Also: *Oracle Database Data Warehousing Guide* for information on initializing and tuning parameters for parallel execution

PARALLEL_MAX_SERVERS

Property	Description
Parameter type	Integer
Default value	Derived from the values of CPU_COUNT, PARALLEL_THREADS_PER_CPU, and PGA_AGGREGATE_TARGET
Modifiable	ALTER SYSTEM
Range of values	0 to 3599
Real Application Clusters	Multiple instances must have the same value.

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

PARALLEL_MAX_SERVERS specifies the maximum number of parallel execution processes and parallel recovery processes for an instance. As demand increases, Oracle increases the number of processes from the number created at instance startup up to this value.

If you set this parameter too low, some queries may not have a parallel execution process available to them during query processing. If you set it too high, memory resource shortages may occur during peak periods, which can degrade performance.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PARALLEL_MIN_PERCENT

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SESSION
Range of values	0 to 100

Property	Description
Real Application Clusters	Multiple instances can have different values.

PARALLEL_MIN_PERCENT operates in conjunction with PARALLEL_MAX_SERVERS and PARALLEL_MIN_SERVERS. It lets you specify the minimum percentage of parallel execution processes (of the value of PARALLEL_MAX_SERVERS) required for parallel execution. Setting this parameter ensures that parallel operations will not execute sequentially unless adequate resources are available. The default value of 0 means that no minimum percentage of processes has been set.

Consider the following settings:

```
PARALLEL_MIN_PERCENT = 50
PARALLEL_MIN_SERVERS = 5
PARALLEL_MAX_SERVERS = 10
```

If 8 of the 10 parallel execution processes are busy, only 2 processes are available. If you then request a query with a degree of parallelism of 8, the minimum 50% will not be met.

You can use this parameter in conjunction with PARALLEL_ADAPTIVE_MULTI_USER. In a multi-user environment, an individual user or application can set PARALLEL_MIN_PERCENT to a minimum value until sufficient resources are available on the system and an acceptable degree of parallelism is returned.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution
- "PARALLEL_MAX_SERVERS" on page 1-119, "PARALLEL_MIN_SERVERS" on page 1-120, and "PARALLEL_ADAPTIVE_MULTI_USER" on page 1-116

PARALLEL_MIN_SERVERS

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to value of PARALLEL_MAX_SERVERS
Real Application Clusters	Multiple instances can have different values.

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

PARALLEL_MIN_SERVERS specifies the minimum number of parallel execution processes for the instance. This value is the number of parallel execution processes Oracle creates when the instance is started.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PARALLEL_THREADS_PER_CPU

Property	Description
Parameter type	Integer
Default value	Operating system-dependent, usually 2
Modifiable	`ALTER SYSTEM`
Range of values	Any nonzero number

Note: This parameter applies to parallel execution in exclusive mode as well as in a Real Application Clusters environment.

`PARALLEL_THREADS_PER_CPU` specifies the default degree of parallelism for the instance and determines the parallel adaptive and load balancing algorithms. The parameter describes the number of parallel execution processes or **threads** that a CPU can handle during parallel execution.

The default is platform-dependent and is adequate in most cases. You should decrease the value of this parameter if the machine appears to be overloaded when a representative parallel query is executed. You should increase the value if the system is I/O bound.

See Also: *Oracle Database Performance Tuning Guide* for more information on tuning parallel execution

PGA_AGGREGATE_TARGET

Property	Description
Parameter type	Big integer
Syntax	`PGA_AGGREGATE_TARGET =` *`integer`* `[K | M | G]`
Default value	10 MB or 20% of the size of the SGA, whichever is greater
Modifiable	`ALTER SYSTEM`
Range of values	Minimum: 10 MB Maximum: 4096 GB - 1
Basic	Yes

`PGA_AGGREGATE_TARGET` specifies the target aggregate PGA memory available to all server processes attached to the instance.

Setting `PGA_AGGREGATE_TARGET` to a nonzero value has the effect of automatically setting the `WORKAREA_SIZE_POLICY` parameter to `AUTO`. This means that SQL working areas used by memory-intensive SQL operators (such as sort, group-by, hash-join, bitmap merge, and bitmap create) will be automatically sized. A nonzero value for this parameter is the default since, unless you specify otherwise, Oracle sets it to 20% of the SGA or 10 MB, whichever is greater.

Setting PGA_AGGREGATE_TARGET to 0 automatically sets the WORKAREA_SIZE_POLICY parameter to MANUAL. This means that SQL workareas are sized using the *_AREA_SIZE parameters.

Oracle attempts to keep the amount of private memory below the target specified by this parameter by adapting the size of the work areas to private memory. When increasing the value of this parameter, you indirectly increase the memory allotted to work areas. Consequently, more memory-intensive operations are able to run fully in memory and less will work their way over to disk.

When setting this parameter, you should examine the total memory on your system that is available to the Oracle instance and subtract the SGA. You can assign the remaining memory to PGA_AGGREGATE_TARGET.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about setting and tuning the PGA_AGGREGATE_TARGET parameter

PLSCOPE_SETTINGS

Property	Description
Parameter type	String
Syntax	PLSCOPE_SETTINGS = IDENTIFIERS:{ NONE | ALL }
Default value	IDENTIFIERS:NONE
Modifiable	ALTER SESSION, ALTER SYSTEM
Basic	No

PLSCOPE_SETTINGS controls the compile time collection, cross reference, and storage of PL/SQL source code identifier data.

Values:

- IDENTIFIERS:NONE

  Disables collection of identifier data. This is the default.

- IDENTIFIERS:ALL

  Enables the collection of all source code identifier data.

PLSCOPE_SETTINGS can be set on a session, system, or per-library unit (ALTER COMPILE) basis. The current setting of PLSCOPE_SETTINGS for any library unit can be attained by querying the *_PLSQL_OBJECT_SETTINGS views. Any identifier data collected by setting this parameter can be accessed using the *_IDENTIFIERS views.

> **See Also:**
>
> - "ALL_PLSQL_OBJECT_SETTINGS" on page 3-10, "DBA_PLSQL_OBJECT_SETTINGS" on page 5-8, and "USER_PLSQL_OBJECT_SETTINGS" on page 5-121
> - "ALL_IDENTIFIERS" on page 2-57, "DBA_IDENTIFIERS" on page 4-135, and "USER_IDENTIFIERS" on page 5-111

PLSQL_CCFLAGS

Property	Description
Parameter type	String
Syntax	PLSQL_CCFLAGS = '<*v1*>:<*c1*>,<*v2*>:<*c2*>,...,<*vn*>:<*cn*>'
Default value	Empty string
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	Any string literal that satisfies the internal syntax

Examples	`ALTER SESSION SET PLSQL_CCFLAGS = 'DeBug:TruE';`
	`ALTER SESSION SET PLSQL_CCFLAGS = 'debug:TRUE';`

`PLSQL_CCFLAGS` provides a mechanism that allows PL/SQL programmers to control conditional compilation of each PL/SQL library unit independently.

Values:

- <*vi*> has the form of an unquoted PL/SQL identifier. It is unrestricted and can be a reserved word or a keyword. The text is insensitive to case. Each one is known as a flag or flag name. Each <*vi*> can occur more than once in the string, each occurrence can have a different flag value, and the flag values can be of different kinds.
- <*ci*> is one of the following: a PL/SQL boolean literal, a `PLS_INTEGER` literal, or the literal NULL. The text is insensitive to case. Each one is known as a flag value and corresponds to a flag name.

You can define any allowable value for `PLSQL_CCFLAGS`. However, Oracle recommends that this parameter be used for controlling the conditional compilation of debugging or tracing code. It is recommended that the following identifiers not be used as flag name values:

- Names of Oracle parameters (for example, `NLS_LENGTH_SEMANTICS`)
- Identifiers with any of the following prefixes: `PLS_`, `PLSQL_`, `PLSCC_`, `ORA_`, `ORACLE_`, `DBMS_`, `SYS_`

PLSQL_CODE_TYPE

Property	Description
Parameter type	String
Syntax	`PLSQL_CODE_TYPE = { INTERPRETED | NATIVE }`
Default value	`INTERPRETED`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`PLSQL_CODE_TYPE` specifies the compilation mode for PL/SQL library units.

Values:

- `INTERPRETED`

  PL/SQL library units will be compiled to PL/SQL bytecode format. Such modules are executed by the PL/SQL interpreter engine.

- NATIVE

  PL/SQL library units (with the possible exception of top-level anonymous PL/SQL blocks) will be compiled to native (machine) code. Such modules will be executed natively without incurring any interpreter overhead.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

If a PL/SQL library unit is compiled native, all subsequent automatic recompilations of that library unit will use native compilation.

PLSQL_DEBUG

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true | false`

> **Note:** The `PLSQL_DEBUG` parameter is deprecated. It is retained for backward compatibility only.

`PLSQL_DEBUG` specifies whether or not PL/SQL library units will be compiled for debugging.

Values:

- `true`

  PL/SQL library units will be compiled for debugging

- `false`

  PL/SQL library units will be compiled for normal execution

When `PLSQL_DEBUG` is set to `true`, PL/SQL library units are always compiled `INTERPRETED` in order to be debuggable.

When the value of this parameter is changed, it has no effect on PL/SQL library units that have already been compiled. The value of this parameter is stored persistently with each library unit.

PLSQL_NATIVE_LIBRARY_DIR

Property	Description
Parameter type	String
Syntax	`PLSQL_NATIVE_LIBRARY_DIR =` *`directory`*
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Range of values	Any valid directory path

PLSQL_NATIVE_LIBRARY_DIR no longer has any effect, and will be made obsolete in a future release of the Oracle Database.

PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to 2^{32} - 1 (max value represented by 32 bits)

PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT no longer has any effect, and will be made obsolete in a future release of the Oracle Database.

PLSQL_OPTIMIZE_LEVEL

Property	Description
Parameter type	Integer
Default value	2
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	0 to 3

PLSQL_OPTIMIZE_LEVEL specifies the optimization level that will be used to compile PL/SQL library units. The higher the setting of this parameter, the more effort the compiler makes to optimize PL/SQL library units.

Values:

- 0

  Maintains the evaluation order and hence the pattern of side effects, exceptions, and package initializations of Oracle9*i* and earlier releases. Also removes the new semantic identity of BINARY_INTEGER and PLS_INTEGER and restores the earlier rules for the evaluation of integer expressions. Although code will run somewhat faster than it did in Oracle9*i*, use of level 0 will forfeit most of the performance gains of PL/SQL in Oracle Database 10*g*.

- 1

  Applies a wide range of optimizations to PL/SQL programs including the elimination of unnecessary computations and exceptions, but generally does not move source code out of its original source order.

- 2

  Applies a wide range of modern optimization techniques beyond those of level 1 including changes which may move source code relatively far from its original location.

- 3

  Applies a wide range of optimization techniques beyond those of level 2, automatically including techniques not specifically requested.

Generally, setting this parameter to `2` pays off in better execution performance. If, however, the compiler runs slowly on a particular source module or if optimization does not make sense for some reason (for example, during rapid turnaround development), then setting this parameter to `1` will result in almost as good a compilation with less use of compile-time resources.

The value of this parameter is stored persistently with the library unit.

PLSQL_V2_COMPATIBILITY

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`true	false`

Note: The `PLSQL_V2_COMPATIBILITY` parameter is deprecated. It is retained for backward compatibility only.

PL/SQL Version 2 allows some abnormal behavior that Version 8 disallows. If you want to retain that behavior for backward compatibility, set `PLSQL_V2_COMPATIBILITY` to `true`. If you set it to `false`, then PL/SQL Version 8 behavior is enforced and Version 2 behavior is not allowed.

See Also: *Oracle Database PL/SQL Language Reference* for a description of the differences between PL/SQL Version 2 and Version 8, and for more information on setting this parameter

PLSQL_WARNINGS

Property	Description
Parameter type	String

Syntax	`PLSQL_WARNINGS = '`*`value_clause`*`' [, '`*`value_clause`*`' ] ...`
	value_clause::=
	`{ ENABLE
		DISABLE	ERROR }:`
			`{ ALL`
			`| SEVERE`
			`| INFORMATIONAL`
			`| PERFORMANCE`
			`| {` *`integer`*
			`| (`*`integer`*` [, `*`integer`*` ] ...)`
			`}`
			`}`

Default value	`'DISABLE:ALL'`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

Property	Description

Examples	`PLSQL_WARNINGS = 'ENABLE:SEVERE', 'DISABLE:INFORMATIONAL';`
	`PLSQL_WARNINGS = 'DISABLE:ALL';`
	`PLSQL_WARNINGS = 'DISABLE:5000', 'ENABLE:5001', 'ERROR:5002';`
	`PLSQL_WARNINGS = 'ENABLE:(5000,5001,5002)', 'DISABLE:(6000,6001)';`

`PLSQL_WARNINGS` enables or disables the reporting of warning messages by the PL/SQL compiler, and specifies which warning messages to show as errors.

value_clause

Multiple value clauses may be specified, enclosed in quotes and separated by commas. Each value clause is composed of a qualifier, a colon (**:**), and a modifier.

Qualifier values:

- `ENABLE`

  Enable a specific warning or a set of warnings

- `DISABLE`

  Disable a specific warning or a set of warnings

- `ERROR`

  Treat a specific warning or a set of warnings as errors

Modifier values:

- `ALL`

  Apply the qualifier to all warning messages

- `SEVERE`

  Apply the qualifier to only those warning messages in the `SEVERE` category

- `INFORMATIONAL`

  Apply the qualifier to only those warning messages in the `INFORMATIONAL` category

- `PERFORMANCE`

  Apply the qualifier to only those warning messages in the `PERFORMANCE` category

PRE_PAGE_SGA

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

`PRE_PAGE_SGA` determines whether Oracle reads the entire SGA into memory at instance startup. Operating system page table entries are then prebuilt for each page of the SGA. This setting can increase the amount of time necessary for instance startup, but it is likely to decrease the amount of time necessary for Oracle to reach its full performance capacity after startup.

> **Note:** This setting does not prevent your operating system from paging or swapping the SGA after it is initially read into memory.

`PRE_PAGE_SGA` can increase the process startup duration, because every process that starts must access every page in the SGA. The cost of this strategy is fixed; however, you might simply determine that 20,000 pages must be touched every time a process starts. This approach can be useful with some applications, but not with all applications. Overhead can be significant if your system frequently creates and destroys processes by, for example, continually logging on and logging off.

The advantage that `PRE_PAGE_SGA` can afford depends on page size. For example, if the SGA is 80 MB in size and the page size is 4 KB, then 20,000 pages must be touched to refresh the SGA (80,000/4 = 20,000).

If the system permits you to set a 4 MB page size, then only 20 pages must be touched to refresh the SGA (80,000/4,000 = 20). The page size is operating system-specific and generally cannot be changed. Some operating systems, however, have a special implementation for shared memory whereby you can change the page size.

PROCESSES

Property	Description
Parameter type	Integer
Default value	100
Modifiable	No
Range of values	6 to operating system dependent
Basic	Yes
Real Application Clusters	Multiple instances can have different values.

`PROCESSES` specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value should allow for all background processes such as locks, job queue processes, and parallel execution processes.

The default values of the `SESSIONS` and `TRANSACTIONS` parameters are derived from this parameter. Therefore, if you change the value of `PROCESSES`, you should evaluate whether to adjust the values of those derived parameters.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter in exclusive mode
- *Oracle Real Application Clusters Administration and Deployment Guide* for information on setting this parameter in a Real Application Clusters environment
- Your operating system-specific Oracle documentation for the range of values

QUERY_REWRITE_ENABLED

Property	Description
Parameter type	String
Syntax	`QUERY_REWRITE_ENABLED = { false | true | force }`
Default value	If `OPTIMIZER_FEATURES_ENABLE` is set to `10.0.0` or higher, then `true` If `OPTIMIZER_FEATURES_ENABLE` is set to `9.2.0` or lower, then `false`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Real Application Clusters	Multiple instances can have different values.

`QUERY_REWRITE_ENABLED` allows you to enable or disable query rewriting globally for the database.

Values:

- `false`

  Oracle does not use rewrite.

- `true`

  Oracle costs the query with rewrite and without rewrite and chooses the method with the lower cost.

- `force`

  Oracle always uses rewrite and does not evaluate the cost before doing so. Use `force` when you know that the query will always benefit from rewrite and when reduction in compile time is important.

To take advantage of query rewrite for a particular materialized view, you must enable query rewrite for that materialized view, and you must enable cost-based optimization.

See Also:

- *Oracle Database Data Warehousing Guide* for information on query rewrite of materialized views
- *Oracle Database Performance Tuning Guide* and "OPTIMIZER_MODE" on page 1-112 for information on cost-based optimization

QUERY_REWRITE_INTEGRITY

Property	Description
Parameter type	String
Syntax	`QUERY_REWRITE_INTEGRITY = { enforced | trusted | stale_tolerated }`
Default value	`enforced`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Real Application Clusters	Multiple instances can have different values.

QUERY_REWRITE_INTEGRITY determines the degree to which Oracle must enforce query rewriting. At the safest level, Oracle does not use query rewrite transformations that rely on unenforced relationships.

Values:

- `enforced`

  Oracle enforces and guarantees consistency and integrity.

- `trusted`

  Oracle allows rewrites using relationships that have been declared, but that are not enforced by Oracle.

- `stale_tolerated`

  Oracle allows rewrites using unenforced relationships. Materialized views are eligible for rewrite even if they are known to be inconsistent with the underlying detail data.

RDBMS_SERVER_DN

Property	Description
Parameter type	X.500 Distinguished Name
Default value	There is no default value.
Modifiable	No
Range of values	All X.500 Distinguished Name format values

RDBMS_SERVER_DN specifies the Distinguished Name (DN) of the Oracle server. It is used for retrieving Enterprise Roles from an enterprise directory service.

If you do not want to use a directory for enterprise user and privilege management, but prefer to use SSL authentication alone, do not set this parameter.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide* for more information on enterprise roles and the enterprise directory service

READ_ONLY_OPEN_DELAYED

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

READ_ONLY_OPEN_DELAYED determines when datafiles in read-only tablespaces are accessed.

Values:

- `true`

The datafiles are accessed for the first time only when an attempt is made to read data stored within them.

- `false`

  The datafiles are accessed at database open time.

You can use this parameter to speed up some operations (primarily opening the database) for very large databases when substantial portions of the database are stored in read-only tablespaces. Consider setting this parameter to `true` for such databases, especially if portions of the read-only data are stored on slow-access devices or hierarchical storage.

> **See Also:** *Oracle Database Administrator's Guide* for information on the consequences of delaying access of datafiles in read-only tablespaces

RECOVERY_PARALLELISM

Property	Description
Parameter type	Integer
Default value	Operating system-dependent
Modifiable	No
Range of values	Operating system-dependent, but cannot exceed `PARALLEL_MAX_SERVERS`

`RECOVERY_PARALLELISM` specifies the number of processes to participate in instance or crash recovery. A value of `0` or `1` indicates that recovery is to be performed serially by one process.

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for information on setting this parameter in exclusive mode
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information on setting this parameter in a Real Application Clusters environment

RECYCLEBIN

Property	Description
Parameter type	String
Syntax	`RECYCLEBIN = { on | off }`
Default value	`on`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM ... DEFERRED`
Basic	No

`RECYCLEBIN` is used to control whether the Flashback Drop capability is turned on or off. If the parameter is set to `off`, then dropped tables do not go into the recycle bin. If this parameter is set to on, then dropped tables go into the recycle bin and can be recovered.

REDO_TRANSPORT_USER

Property	Description
Parameter type	String
Syntax	`REDO_TRANSPORT_USER =` *`user_name`*
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Basic	No

`REDO_TRANSPORT_USER` specifies the name of the user whose password verifier is used when a remote login password file is used for redo transport authentication. This user must have `SYSOPER` privileges and must have the same password in the database that initiates the redo transport session and in the database that is the target of the redo transport session.

If this parameter is not specified, then the password verifier of the `SYS` user will be used when a remote login password file is used for redo transport authentication.

REMOTE_DEPENDENCIES_MODE

Property	Description
Parameter type	String
Syntax	`REMOTE_DEPENDENCIES_MODE = { TIMESTAMP | SIGNATURE }`
Default value	`TIMESTAMP`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`REMOTE_DEPENDENCIES_MODE` specifies how Oracle should handle dependencies upon remote PL/SQL stored procedures.

Values:

- `TIMESTAMP`

  The client running the procedure compares the timestamp recorded on the server-side procedure with the current timestamp of the local procedure and executes the procedure only if the timestamps match.

- `SIGNATURE`

  Oracle allows the procedure to execute as long as the signatures are considered safe. This setting allows client PL/SQL applications to be run without recompilation.

> **See Also:** *Oracle Database Advanced Application Developer's Guide* for information about the consequences of the settings of this parameter

REMOTE_LISTENER

Property	Description
Parameter type	String
Syntax	`REMOTE_LISTENER = network_name`
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Basic	Yes

`REMOTE_LISTENER` specifies a network name that resolves to an address or address list of Oracle Net remote listeners (that is, listeners that are not running on the same machine as this instance). The address or address list is specified in the `TNSNAMES.ORA` file or other address repository as configured for your system.

See Also:

- *Oracle Database Concepts* for more information about instances, listener processes, and dispatcher processes
- *Oracle Database Net Services Administrator's Guide* and your operating system-specific Oracle documentation for more information about specifying network addresses for the protocols on your system

REMOTE_LOGIN_PASSWORDFILE

Property	Description
Parameter type	String
Syntax	`REMOTE_LOGIN_PASSWORDFILE = { shared | exclusive | none }`
Default value	`exclusive`
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

`REMOTE_LOGIN_PASSWORDFILE` specifies whether Oracle checks for a password file.

Values:

- `shared`

  One or more databases can use the password file. The password file can contain `SYS` as well as non-`SYS` users.

- `exclusive`

  The password file can be used by only one database. The password file can contain `SYS` as well as non-`SYS` users.

- `none`

Oracle ignores any password file. Therefore, privileged users must be authenticated by the operating system.

> **Note:** When `REMOTE_LOGIN_PASSWORDFILE` is set to either `exclusive` or `shared`, but the password file does not exist, then the behavior is the same as setting `REMOTE_LOGIN_PASSWORDFILE` to `none`.

REMOTE_OS_AUTHENT

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

`REMOTE_OS_AUTHENT` specifies whether remote clients will be authenticated with the value of the `OS_AUTHENT_PREFIX` parameter.

> **Note:** The `REMOTE_OS_AUTHENT` parameter is deprecated. It is retained for backward compatibility only.

See Also:

- *Oracle Database Advanced Security Administrator's Guide* for more information on setting this parameter
- "OS_AUTHENT_PREFIX" on page 1-115

REMOTE_OS_ROLES

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true` | `false`

`REMOTE_OS_ROLES` specifies whether operating system roles are allowed for remote clients. The default value, `false`, causes Oracle to identify and manage roles for remote clients.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- "OS_ROLES" on page 1-116

REPLICATION_DEPENDENCY_TRACKING

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	No
Range of values	`true` | `false`

`REPLICATION_DEPENDENCY_TRACKING` enables or disables dependency tracking for read/write operations to the database. Dependency tracking is essential for propagating changes in a replicated environment in parallel.

Values:

- `TRUE`

  Enables dependency tracking.

- `FALSE`

  Allows read/write operations to the database to run faster, but does not produce dependency information for Oracle to perform parallel propagation.

  > **Note:** Do not specify this value unless you are sure that your application will not perform any read/write operations to the replicated tables.

  > **See Also:** *Oracle Database Advanced Replication* for more information on parallel propagation dependency tracking

RESOURCE_LIMIT

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	`ALTER SYSTEM`
Range of values	`true` | `false`

`RESOURCE_LIMIT` determines whether resource limits are enforced in database profiles.

Values:

- `TRUE`

  Enables the enforcement of resource limits

- `FALSE`

  Disables the enforcement of resource limits

See Also: *Oracle Database Administrator's Guide* and *Oracle Database SQL Language Reference* for more information on setting resource limits for profiles

RESOURCE_MANAGER_CPU_ALLOCATION

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SYSTEM`
Range of values	`0` to operating system-specific
Basic	No

`RESOURCE_MANAGER_CPU_ALLOCATION` specifies the number of CPUs that the Resource Manager should utilize. The Resource Manager controls how a system's CPUs are utilized by its database's sessions.

The Resource Manager schedules database sessions on the CPUs according to a Resource Plan that has been configured and enabled by the DBA. Normally, the Resource Manager schedules enough database sessions to keep all CPUs utilized. However, in some scenarios, a DBA may only want to schedule enough database sessions to keep a subset of the CPUs utilized.

RESOURCE_MANAGER_PLAN

Property	Description
Parameter type	String
Syntax	`RESOURCE_MANAGER_PLAN =` *`plan_name`*
Default value	There is no default value.
Modifiable	`ALTER SYSTEM`
Range of values	Any valid character string

`RESOURCE_MANAGER_PLAN` specifies the top-level resource plan to use for an instance. The resource manager will load this top-level plan along with all its descendants (subplans, directives, and consumer groups). If you do not specify this parameter, the resource manager is off by default.

You can change the setting of this parameter using the `ALTER SYSTEM` statement to turn on the resource manager (if it was previously off) or to turn off the resource manager or change the current plan (if it was previously on). If you specify a plan that does not exist in the data dictionary, Oracle returns an error message.

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans
- *Oracle Database PL/SQL Packages and Types Reference* for information on the DBMS_RESOURCE_MANAGER and DBMS_RESOURCE_MANAGER_PRIVS packages
- "DBA_RSRC_PLANS" on page 5-27, "DBA_RSRC_PLAN_DIRECTIVES" on page 5-25, and the various V$RSRC_* dynamic performance views in Part III for information on existing resource plans

RESULT_CACHE_MAX_RESULT

Property	Description
Parameter type	Integer
Default value	5 percent
Modifiable	ALTER SYSTEM
Range of values	0 to 100
Basic	No

RESULT_CACHE_MAX_RESULT specifies the percentage of RESULT_CACHE_MAX_SIZE that any single result can use.

RESULT_CACHE_MAX_SIZE

Property	Description
Parameter type	Big integer
Syntax	RESULT_CACHE_MAX_SIZE = *integer* [K | M | G]
Default value	Derived from the values of SHARED_POOL_SIZE, SGA_TARGET, and MEMORY_TARGET
Modifiable	ALTER SYSTEM
Range of values	0 to operating system-dependent
Basic	No
Real Application Clusters	you must either set this parameter to 0 on all instances to disable the result cache, or use a nonzero value on all instances. Disabling the result cache on some instances may lead to incorrect results.

RESULT_CACHE_MAX_SIZE specifies the maximum amount of SGA memory (in bytes) that can be used by the Result Cache. Values of this parameter greater than 0 are rounded up to the next multiple of 32 KB. If the value of this parameter is 0, then the feature is disabled.

RESULT_CACHE_MODE

Property	Description
Parameter type	String
Syntax	`RESULT_CACHE_MODE = { MANUAL | FORCE }`
Default value	`MANUAL`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`RESULT_CACHE_MODE` specifies when a `ResultCache` operator is spliced into a query's execution plan.

Values:

- `MANUAL`

  The `ResultCache` operator is added only when the query is annotated (that is, hints).

- `FORCE`

  The `ResultCache` operator is added to the root of all `SELECT` statements (provided that it is valid to do so).

For the `FORCE` setting, if the statement contains a `NO_RESULT_CACHE` hint, then the hint takes precedence over the parameter setting.

RESULT_CACHE_REMOTE_EXPIRATION

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Range of values	`0` to operating system-dependent
Basic	No

`RESULT_CACHE_REMOTE_EXPIRATION` specifies the number of minutes that a result using a remote object is allowed to remain valid. Setting this parameter to `0` implies that results using remote objects should not be cached. Ssetting this parameter to a nonzero value can produce stale answers (for example, if the remote table used by a result is modified at the remote database).

RESUMABLE_TIMEOUT

Property	Description
Parameter type	Integer
Default value	`0` (seconds)
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

Property	Description
Range of values	0 to 2^{31} - 1 (in seconds)
Real Application Clusters	Multiple instances can have different values.

RESUMABLE_TIMEOUT enables or disables resumable statements and specifies resumable timeout at the system level.

> **See Also:** *Oracle Database Administrator's Guide* for more information about enabling resumable space allocation, what conditions are correctable, and what statements can be made resumable

ROLLBACK_SEGMENTS

Property	Description
Parameter type	String
Syntax	ROLLBACK_SEGMENTS = (segment_name [, segment_name] ...)
Default value	If you do not specify this parameter, the instance uses public rollback segments by default, unless the UNDO_MANAGEMENT initialization parameter is set to AUTO. In that case, the ROLLBACK_SEGMENTS parameter is ignored and automatic undo management is used.
Modifiable	No
Range of values	Any rollback segment names listed in DBA_ROLLBACK_SEGS except SYSTEM
Basic	Yes
Real Application Clusters	Multiple instances must have different values.

ROLLBACK_SEGMENTS allocates one or more rollback segments by name to this instance. If you set this parameter, the instance acquires all of the rollback segments named in this parameter, even if the number of rollback segments exceeds the minimum number required by the instance (calculated as TRANSACTIONS / TRANSACTIONS_PER_ROLLBACK_SEGMENT).

You cannot change the value of this parameter dynamically, but you can change its value and then restart the instance. Although this parameter usually specifies private rollback segments, it can also specify public rollback segments if they are not already in use.

To find the name, segment ID number, and status of each rollback segment in the database, query the data dictionary view DBA_ROLLBACK_SEGS.

When UNDO_MANAGEMENT is set to AUTO, ROLLBACK_SEGMENTS is ignored.

See Also:

- *Oracle Database Administrator's Guide* for more information on setting this parameter
- *Oracle Real Application Clusters Administration and Deployment Guide* for information on setting this parameter in a Real Application Clusters environment
- "DBA_ROLLBACK_SEGS" on page 5-22

SEC_CASE_SENSITIVE_LOGON

Property	Description
Parameter type	Boolean
Default value	`true`
Modifiable	`ALTER SYSTEM`
Range of values	`true | false`
Basic	No

`SEC_CASE_SENSITIVE_LOGON` enables or disables password case sensitivity in the database.

Values:

- `true`

  Database logon passwords are case sensitive.

- `false`

  Database logon passwords are not case sensitive.

SEC_MAX_FAILED_LOGIN_ATTEMPTS

Property	Description
Parameter type	Integer
Default value	`10`
Modifiable	No
Range of values	1 to unlimited
Basic	No

`SEC_MAX_FAILED_LOGIN_ATTEMPTS` specifies the number of authentication attempts that can be made by a client on a connection to the server process. After the specified number of failure attempts, the connection will be automatically dropped by the server process.

SEC_PROTOCOL_ERROR_FURTHER_ACTION

Property	Description
Parameter type	String
Syntax	`SEC_PROTOCOL_ERROR_FURTHER_ACTION = { CONTINUE | (DELAY,`*`integer`*`) | (DROP,`*`integer`*`) }`
Default value	`CONTINUE`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`SEC_PROTOCOL_ERROR_FURTHER_ACTION` specifies the further execution of a server process when receiving bad packets from a possibly malicious client.

Values:

- `CONTINUE`

  The server process continues execution. The database server may be subject to a Denial of Service (DoS) if bad packets continue to be sent by a malicious client.

- `(DELAY,`*`integer`*`)`

  The client experiences a delay of *integer* seconds before the server process accepts the next request from the same client connection. Malicious clients are prevented from excessive consumption of server resources while legitimate clients experience a degradation in performance but can continue to function.

- `(DROP,`*`integer`*`)`

  The server forcefully terminates the client connection after *integer* bad packets. The server protects itself at the expense of the client (for example, a client transaction may be lost). The client may reconnect and attempt the same operation.

SEC_PROTOCOL_ERROR_TRACE_ACTION

Property	Description
Parameter type	String
Syntax	`SEC_PROTOCOL_ERROR_TRACE_ACTION = { NONE | TRACE | LOG | ALERT }`
Default value	`TRACE`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`SEC_PROTOCOL_ERROR_TRACE_ACTION` specifies the action that the database should take when bad packets are received from a possibly malicious client.

Values:

- `NONE`

  The database server ignores the bad packets and does not generate any trace files or log messages.

- `TRACE`

A detailed trace file is generated when bad packets are received, which can be used to debug any problems in client/server communication.

- `LOG`

  A minimal log message is printed in the alert logfile and in the server trace file. A minimal amount of disk space is used.

- `ALERT`

  An alert message is sent to a DBA or monitoring console.

SEC_RETURN_SERVER_RELEASE_BANNER

Property	Description
Parameter type	Boolean
Default value	`false`
Modifiable	No
Range of values	`true | false`
Basic	No

`SEC_RETURN_SERVER_RELEASE_BANNER` specifies whether or not the server returns complete database software information to clients.

Values:

- `true`

  Returns complete database version information to the client.

- `false`

  Returns a generic version string to the client.

SERIAL_REUSE

Property	Description
Parameter type	String
Syntax	`SERIAL_REUSE = { disable | all | select | dml | plsql | force }`
Default value	`disable`
Modifiable	No
Basic	No

Note: The `SERIAL_REUSE` parameter is deprecated. It is retained for backward compatibility only.

`SERIAL_REUSE` specifies which types of cursors make use of the serial-reusable memory feature. This feature allocates private cursor memory in the SGA so that it can be reused (serially, not concurrently) by sessions executing the same cursor.

Values:

- `disable`

  Disables the option for all SQL statement types. This value overrides any other values included in the list.

- `all`

  Enables the option for both DML and `SELECT` statements. Equivalent to setting `select`, `dml`, and `plsql`.

- `select`

  Enables the option for `SELECT` statements.

- `dml`

  Enables the option for DML statements.

- `plsql`

  Currently has no effect (although PLSQL packages do support the serial-reuse memory option using PLSQL pragmas).

> **Note:** If `CURSOR_SPACE_FOR_TIME` is set to `true`, then the value of `SERIAL_REUSE` is ignored and treated as if it were set to `disable`.

See Also: "CURSOR_SPACE_FOR_TIME" on page 1-38

SERVICE_NAMES

Property	Description
Parameter type	String

Syntax	`SERVICE_NAMES =`
	`   `*`db_service_name`*` [, `*`db_service_name`*` [ ... ] ]`

Default value	`DB_UNIQUE_NAME.DB_DOMAIN` if defined
Modifiable	`ALTER SYSTEM`
Range of values	Any ASCII string or comma-separated list of string names
Basic	No
Real Application Clusters	Do not set the `SERVER_NAMES` parameter for Real Application Clusters (RAC). Instead, define services using Database Configuration Assistant (DBCA) and manage services using Server Control (SRVCTL) utility.

`SERVICE_NAMES` specifies one or more names by which clients can connect to the instance. The instance registers its service names with the listener. When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

You can specify multiple service names in order to distinguish among different uses of the same database. For example:

```
SERVICE_NAMES = sales.acme.com, widgetsales.acme.com
```

You can also use service names to identify a single service that is available from two different databases through the use of replication.

If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the DB_DOMAIN parameter. If DB_DOMAIN is not specified, then no domain will be applied to the non-qualified SERVICE_NAMES values.

See Also:

- *Oracle Database Net Services Administrator's Guide* for more information on this parameter and its settings
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about defining and administering services in a RAC environment
- "DB_DOMAIN" on page 1-45

SESSION_CACHED_CURSORS

Property	Description
Parameter type	Integer
Default value	50
Modifiable	ALTER SESSION
Range of values	0 to operating system-dependent
Real Application Clusters	Multiple instances can have different values.

SESSION_CACHED_CURSORS specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed.

This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are re-executed by a user.

See Also: *Oracle Database Performance Tuning Guide* for more information on setting this parameter and its uses

SESSION_MAX_OPEN_FILES

Property	Description
Parameter type	Integer
Default value	10
Modifiable	No
Range of values	1 to either 50 or the value of MAX_OPEN_FILES defined at the operating system level, whichever is less

SESSION_MAX_OPEN_FILES specifies the maximum number of BFILEs that can be opened in any session. Once this number is reached, subsequent attempts to open more files in the session by using DBMS_LOB.FILEOPEN() or OCILobFileOpen() will fail. The maximum value for this parameter depends on the equivalent parameter defined for the underlying operating system.

See Also:

- *Oracle Database Large Objects Developer's Guide* for information on LOBs in general and BFILEs in particular
- *Oracle Database PL/SQL Packages and Types Reference* for information on the DBMS_LOB.FILEOPEN() procedure
- *Oracle Call Interface Programmer's Guide* for information on the OCILobFileOpen() procedure

SESSIONS

Property	Description
Parameter type	Integer
Default value	Derived: (1.1 * PROCESSES) + 5
Modifiable	No
Range of values	1 to 2^{31}
Basic	Yes

SESSIONS specifies the maximum number of sessions that can be created in the system. Because every login requires a session, this parameter effectively determines the maximum number of concurrent users in the system. You should always set this parameter explicitly to a value equivalent to your estimate of the maximum number of concurrent users, plus the number of background processes, plus approximately 10% for recursive sessions.

Oracle uses the default value of this parameter as its minimum. Values between 1 and the default do not trigger errors, but Oracle ignores them and uses the default instead.

The default values of the ENQUEUE_RESOURCES and TRANSACTIONS parameters are derived from SESSIONS. Therefore, if you increase the value of SESSIONS, you should consider whether to adjust the values of ENQUEUE_RESOURCES and TRANSACTIONS as well. (Note that ENQUEUE_RESOURCES is obsolete as of Oracle Database 10g release 2 (10.2).)

In a shared server environment, the value of PROCESSES can be quite small. Therefore, Oracle recommends that you adjust the value of SESSIONS to approximately 1.1 * total number of *connections*.

See Also:

- *Oracle Database Concepts* for more information on memory structures and processes

SGA_MAX_SIZE

Property	Description
Parameter type	Big integer
Syntax	`SGA_MAX_SIZE = *integer* [K | M | G]`
Default value	Initial size of SGA at startup, dependent on the sizes of different pools in the SGA, such as buffer cache, shared pool, large pool, and so on.
Modifiable	No
Range of values	0 to operating system-dependent

`SGA_MAX_SIZE` specifies the maximum size of the SGA for the lifetime of the instance.

SGA_TARGET

Property	Description
Parameter type	Big integer
Syntax	`SGA_TARGET = *integer* [K | M | G]`
Default value	`0` (SGA autotuning is disabled)
Modifiable	`ALTER SYSTEM`
Range of values	64 MB to operating system-dependent
Basic	Yes

`SGA_TARGET` specifies the total size of all SGA components. If `SGA_TARGET` is specified, then the following memory pools are automatically sized:

- Buffer cache (`DB_CACHE_SIZE`)
- Shared pool (`SHARED_POOL_SIZE`)
- Large pool (`LARGE_POOL_SIZE`)
- Java pool (`JAVA_POOL_SIZE`)
- Streams pool (`STREAMS_POOL_SIZE`)

If these automatically tuned memory pools are set to nonzero values, then those values are used as minimum levels by Automatic Shared Memory Management. You would set minimum values if an application component needs a minimum amount of memory to function properly.

The following pools are manually sized components and are not affected by Automatic Shared Memory Management:

- Log buffer
- Other buffer caches, such as `KEEP`, `RECYCLE`, and other block sizes
- Fixed SGA and other internal allocations

The memory allocated to these pools is deducted from the total available for `SGA_TARGET` when Automatic Shared Memory Management computes the values of the automatically tuned memory pools.

See Also:

- *Oracle Database Concepts* for information on automatic SGA management
- *Oracle Database Administrator's Guide* for information on managing the SGA

SHADOW_CORE_DUMP

Property	Description
Parameter type	String
Syntax	`SHADOW_CORE_DUMP = { partial | full | none }`
Default value	`partial`
Modifiable	No

`SHADOW_CORE_DUMP` specifies whether Oracle includes the SGA in the core file for foreground (client) processes.

Values:

- `partial`

  Oracle does not include the SGA in the core dump.

- `full`

  Oracle includes the SGA in the core dump.

- `none`

  No core files will be generated for foreground processes.

See Also: "BACKGROUND_CORE_DUMP" on page 1-25

SHARED_MEMORY_ADDRESS

Property	Description
Parameter type	Integer
Default value	`0`
Modifiable	No

`SHARED_MEMORY_ADDRESS` and `HI_SHARED_MEMORY_ADDRESS` specify the starting address at runtime of the system global area (SGA). This parameter is ignored on the many platforms that specify the SGA's starting address at linktime.

Use this parameter to specify the entire address on 32-bit platforms and to specify the low-order 32 bits of a 64-bit address on 64-bit platforms. Use `HI_SHARED_MEMORY_ADDRESS` to specify the high-order 32 bits of a 64-bit address on 64-bit platforms. If both parameters are 0 or unspecified, the SGA address defaults to a platform-specific location.

See Also: "HI_SHARED_MEMORY_ADDRESS" on page 1-68

SHARED_POOL_RESERVED_SIZE

Property	Description
Parameter type	Big integer
Syntax	`SHARED_POOL_RESERVED_SIZE = *integer* [K | M | G]`
Default value	5% of the value of `SHARED_POOL_SIZE`
Modifiable	No
Range of values	Minimum: `5000` Maximum: one half of the value of `SHARED_POOL_SIZE`

`SHARED_POOL_RESERVED_SIZE` specifies (in bytes) the shared pool space that is reserved for large contiguous requests for shared pool memory.

You can use this parameter to avoid performance degradation in the shared pool in situations where pool fragmentation forces Oracle to search for and free chunks of unused pool to satisfy the current request.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "SHARED_POOL_SIZE" on page 1-148

SHARED_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	`SHARED_POOL_SIZE = *integer* [K | M | G]`
Default value	If `SGA_TARGET` is set: If the parameter is not specified, then the default is `0` (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If `SGA_TARGET` is not set (32-bit platforms): 32 MB, rounded up to the nearest granule size. If `SGA_TARGET` is not set (64-bit platforms): 84 MB, rounded up to the nearest granule size. For considerations when dealing with database instances using ASM, see "SHARED_POOL_SIZE and Automatic Storage Management" on page 1-149.
Modifiable	`ALTER SYSTEM`
Range of values	Minimum: the granule size Maximum: operating system-dependent

`SHARED_POOL_SIZE` specifies (in bytes) the size of the shared pool. The shared pool contains shared cursors, stored procedures, control structures, and other structures. If you set `PARALLEL_AUTOMATIC_TUNING` to `false`, then Oracle also allocates parallel execution message buffers from the shared pool. Larger values improve performance in multi-user systems. Smaller values use less memory.

You can monitor utilization of the shared pool by querying the view V$SGASTAT.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Upgrade Guide* for information on parallel execution message buffers
- "PARALLEL_AUTOMATIC_TUNING" on page 1-117 and "V$SGASTAT" on page 8-23

SHARED_POOL_SIZE and Automatic Storage Management

On a database instance using ASM, additional memory is required to store extent maps. As a general guideline, you can aggregate the values from the following queries to obtain current database storage size that is either already on ASM or will be stored in ASM. Then determine the redundancy type that is used (or will be used), and calculate the value for SHARED_POOL_SIZE, using the aggregated value as input.

```
SELECT SUM(BYTES)/(1024*1024*1024) FROM V$DATAFILE;
SELECT SUM(BYTES)/(1024*1024*1024) FROM V$LOGFILE a, V$LOG b
WHERE a.group#=b.group#;
SELECT SUM(BYTES)/(1024*1024*1024) FROM V$TEMPFILE WHERE
status='ONLINE';
```

Additionally, keep the following guidelines in mind:

- For diskgroups using external redundancy:

  (Every 100G of space needs 1M of extra shared pool) + 2M
- For diskgroups using normal redundancy:

  (Every 50G of space needs 1M of extra shared pool) + 4M
- For diskgroups using high redundancy:

  (Every 33G of space needs 1M of extra shared pool) + 6M

SHARED_SERVER_SESSIONS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If SHARED_SERVER_SESSIONS is specified, then it should be less than SESSIONS. If SHARED_SERVER_SESSIONS is not specified, then a shared server session may be created as long as there is a free session slot.

SHARED_SERVER_SESSIONS specifies the total number of shared server sessions to allow. Setting this parameter enables you to reserve user sessions for dedicated servers.

See Also: *Oracle Database Concepts* for more information on sessions

SHARED_SERVERS

Property	Description
Parameter type	Integer
Default value	0, meaning that shared server is not on. If you are using shared server architecture or if the DISPATCHERS parameter is set such that the total number of dispatchers is more than 0, then the default value is 1.
Modifiable	ALTER SYSTEM
Range of values	The value of this parameter should be less than MAX_SHARED_SERVERS. If it is greater than or equal to MAX_SHARED_SERVERS, then the number of servers will not be self-tuned but will remain constant, as specified by SHARED_SERVERS.
Basic	Yes

SHARED_SERVERS specifies the number of server processes that you want to create when an instance is started. If system load decreases, then this minimum number of servers is maintained. Therefore, you should take care not to set SHARED_SERVERS too high at system startup.

> **See Also:** *Oracle Database Administrator's Guide* for more information on setting this parameter

SKIP_UNUSABLE_INDEXES

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true | false

SKIP_UNUSABLE_INDEXES enables or disables the use and reporting of tables with unusable indexes or index partitions.

Values:

- true

  Disables error reporting of indexes and index partitions marked UNUSABLE. This setting allows all operations (inserts, deletes, updates, and selects) on tables with unusable indexes or index partitions.

  > **Note:** If an index is used to enforce a UNIQUE constraint on a table, then allowing insert and update operations on the table might violate the constraint. Therefore, this setting does not disable error reporting for unusable indexes that are unique.

- false

Enables error reporting of indexes marked UNUSABLE. This setting does not allow inserts, deletes, and updates on tables with unusable indexes or index partitions.

SMTP_OUT_SERVER

Property	Description
Parameter type	String

Syntax	`SMTP_OUT_SERVER = server_clause [, server_clause ] ...`
	server_clause::=
	`host_name[:port]`

Default value	There is no default value.
Modifiable	No

`SMTP_OUT_SERVER` specifies the SMTP host and port to which `UTL_MAIL` delivers out-bound E-mail. Multiple servers may be specified, separated by commas.

If the first server in the list is unavailable, then `UTL_MAIL` tries the second server, and so on.

If `SMTP_OUT_SERVER` is not specified, then the SMTP server name defaults to the value of `DB_DOMAIN`, the port number defaults to `25`, and the SMTP domain defaults to the suffix of `DB_DOMAIN`.

> **See Also:** *Oracle Database PL/SQL Packages and Types Reference* for information on the `UTL_MAIL` package

SORT_AREA_RETAINED_SIZE

Property	Description
Parameter type	Integer
Default value	Derived from `SORT_AREA_SIZE`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM ... DEFERRED`
Range of values	From the value equivalent of two database blocks to the value of `SORT_AREA_SIZE`

> **Note:** Oracle does not recommend using the `SORT_AREA_RETAINED_SIZE` parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting `PGA_AGGREGATE_TARGET` instead. `SORT_AREA_RETAINED_SIZE` is retained for backward compatibility.

`SORT_AREA_RETAINED_SIZE` specifies (in bytes) the maximum amount of the user global area (UGA) memory retained after a sort run completes. The retained size controls the size of the read buffer, which Oracle uses to maintain a portion of the sort in memory. This memory is released back to the UGA, not to the operating system, after the last row is fetched from the sort space.

Oracle may allocate multiple sort spaces of this size for each query. Usually, only one or two sorts occur at one time, even for complex queries. In some cases, however, additional concurrent sorts are required, and each sort keeps its own memory area. If the shared server is used, allocation is to the SGA until the value in SORT_AREA_RETAINED_SIZE is reached. The difference between SORT_AREA_RETAINED_SIZE and SORT_AREA_SIZE is allocated to the PGA.

Note: The default value as reflected in the V$PARAMETER dynamic performance view is 0. However, if you do not explicitly set this parameter, Oracle actually uses the value of the SORT_AREA_SIZE parameter.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- "SORT_AREA_SIZE" on page 1-152

SORT_AREA_SIZE

Property	Description
Parameter type	Integer
Default value	65536
Modifiable	ALTER SESSION, ALTER SYSTEM ... DEFERRED
Range of values	Minimum: the value equivalent of six database blocks Maximum: operating system-dependent

Note: Oracle does not recommend using the SORT_AREA_SIZE parameter unless the instance is configured with the shared server option. Oracle recommends that you enable automatic sizing of SQL working areas by setting PGA_AGGREGATE_TARGET instead. SORT_AREA_SIZE is retained for backward compatibility.

SORT_AREA_SIZE specifies (in bytes) the maximum amount of memory Oracle will use for a sort. After the sort is complete, but before the rows are returned, Oracle releases all of the memory allocated for the sort, except the amount specified by the SORT_AREA_RETAINED_SIZE parameter. After the last row is returned, Oracle releases the remainder of the memory.

Increasing SORT_AREA_SIZE size improves the efficiency of large sorts.

Each sort in a query can consume memory up to the amount specified by SORT_AREA_SIZE, and there can be multiple sorts in a query. Also, if a query is executed in parallel, each PQ slave can consume memory up to the amount specified by SORT_AREA_SIZE for each sort it does.

SORT_AREA_SIZE is also used for inserts and updates to bitmap indexes. Setting this value appropriately results in a bitmap segment being updated only once for each DML operation, even if more than one row in that segment changes.

Larger values of SORT_AREA_SIZE permit more sorts to be performed in memory. If more space is required to complete the sort than will fit into the memory provided, then temporary segments on disk are used to hold the intermediate sort runs.

The default is adequate for most OLTP operations. You might want to adjust this parameter for decision support systems, batch jobs, or large CREATE INDEX operations.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter
- *Oracle Database Concepts* for information on logical storage structures such as sort areas
- Your operating system-specific Oracle documentation for the default value on your system
- "SORT_AREA_RETAINED_SIZE" on page 1-151

SPFILE

Property	Description
Parameter type	String
Syntax	SPFILE = *spfile_name*
Default value	*ORACLE_HOME*/dbs/spfile.ora
Modifiable	No
Range of values	Any valid SPFILE
Real Application Clusters	Multiple instances should have the same value.

The value of this parameter is the name of the current server parameter file (SPFILE) in use. This parameter can be defined in a client side PFILE to indicate the name of the server parameter file to use.

When the default server parameter file is used by the server, the value of SPFILE is internally set by the server.

The SPFILE resides in the *ORACLE_HOME*/dbs directory; however, users can place it anywhere on their machine as long as it is specified in an initialization parameter file.

See Also: *Oracle Database Administrator's Guide* for more information about creating the server parameter file

SQL_TRACE

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true | false

SQL_TRACE enables or disables the SQL trace facility. Setting this parameter to true provides information on tuning that you can use to improve performance. You can change the value using the DBMS_SYSTEM package.

Caution: Using this initialization parameter to enable the SQL trace facility for the entire instance can have a severe performance impact. Enable the facility for specific sessions using the ALTER SESSION statement. If you must enable the facility on an entire production environment, then you can minimize performance impact by:

- Maintaining at least 25% idle CPU capacity
- Maintaining adequate disk space for the USER_DUMP_DEST location
- Striping disk space over sufficient disks

See Also: *Oracle Database Performance Tuning Guide* for more information about performance diagnostic tools

Note: The SQL_TRACE parameter is deprecated. Oracle recommends that you use the DBMS_MONITOR and DBMS_SESSION packages instead. SQL_TRACE is retained for backward compatibility only.

SQL92_SECURITY

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No
Range of values	true | false

The SQL92 standards specify that security administrators should be able to require that users have SELECT privilege on a table when executing an UPDATE or DELETE statement that references table column values in a WHERE or SET clause. SQL92_SECURITY specifies whether users must have been granted the SELECT object privilege in order to execute such UPDATE or DELETE statements.

SQLTUNE_CATEGORY

Property	Description
Parameter type	String
Syntax	SQLTUNE_CATEGORY = *category_name*
Default value	DEFAULT
Modifiable	ALTER SESSION, ALTER SYSTEM

SQLTUNE_CATEGORY specifies the category name for use by sessions to qualify the lookup of SQL profiles during SQL compilation.

> **See Also:** *Oracle Database PL/SQL Packages and Types Reference* for information on the DBMS_SQLTUNE package

STANDBY_ARCHIVE_DEST

Property	Description
Parameter type	String
Syntax	STANDBY_ARCHIVE_DEST = *filespec*
Default value	Operating system-specific
Modifiable	ALTER SYSTEM
Range of values	A valid path or device name other than RAW

STANDBY_ARCHIVE_DEST is relevant only for a standby database in managed recovery mode. It specifies the location of archive logs arriving from a primary database. Oracle uses STANDBY_ARCHIVE_DEST and LOG_ARCHIVE_FORMAT to fabricate the fully qualified standby log filenames and stores the filenames in the standby control file.

You can see the value of this parameter by querying the V$ARCHIVE_DEST data dictionary view.

> **See Also:**
>
> - *Oracle Data Guard Concepts and Administration* for more information on setting this parameter and on managed recovery mode in general
> - "LOG_ARCHIVE_DEST" on page 1-80 and "V$ARCHIVE_DEST" on page 6-8

> **Note:** The STANDBY_ARCHIVE_DEST parameter is deprecated. It is retained for backward compatibility only.

STANDBY_FILE_MANAGEMENT

Property	Description
Parameter type	String
Syntax	STANDBY_FILE_MANAGEMENT = { MANUAL | AUTO }
Default value	MANUAL
Modifiable	ALTER SYSTEM

STANDBY_FILE_MANAGEMENT enables or disables automatic standby file management. When automatic standby file management is enabled, operating system file additions and deletions on the primary database are replicated on the standby database. STANDBY_FILE_MANAGEMENT is only applicable to Physical standby databases.

Values:

- `MANUAL`

  disables automatic standby file management

- `AUTO`

  enables automatic standby file management

Setting `STANDBY_FILE_MANAGEMENT` to `AUTO` causes Oracle to automatically create files on the standby database and, in some cases, overwrite existing files. Care must be taken when setting `STANDBY_FILE_MANAGEMENT` and `DB_FILE_NAME_CONVERT` so that existing standby files will not be accidentally overwritten.

If the standby database is on the same system as the primary database, then ensure that the primary and standby systems do not point to the same files.

> **See Also:** *Oracle Data Guard Concepts and Administration* for more information about setting this parameter

STAR_TRANSFORMATION_ENABLED

Property	Description
Parameter type	String
Syntax	`STAR_TRANSFORMATION_ENABLED = { FALSE | TRUE | TEMP_DISABLE }`
Default value	`FALSE`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	Yes

`STAR_TRANSFORMATION_ENABLED` determines whether a cost-based query transformation will be applied to star queries.

Values:

- `FALSE`

  The transformation will not be applied.

- `TRUE`

  The optimizer will consider performing a cost-based query transformation on the star query.

- `TEMP_DISABLE`

  The optimizer will consider performing a cost-based query transformation on the star query but will not use temporary tables in the star transformation.

> **See Also:**
>
> - *Oracle Database Concepts* for information on star queries
> - *Oracle Database Performance Tuning Guide* for information on enabling star query

STATISTICS_LEVEL

Property	Description
Parameter type	String
Syntax	`STATISTICS_LEVEL = { ALL | TYPICAL | BASIC }`
Default value	`TYPICAL`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`STATISTICS_LEVEL` specifies the level of collection for database and operating system statistics. The Oracle Database collects these statistics for a variety of purposes, including making self-management decisions.

The default setting of `TYPICAL` ensures collection of all major statistics required for database self-management functionality and provides best overall performance. The default value should be adequate for most environments.

When the `STATISTICS_LEVEL` parameter is set to `ALL`, additional statistics are added to the set of statistics collected with the `TYPICAL` setting. The additional statistics are timed OS statistics and plan execution statistics.

Setting the `STATISTICS_LEVEL` parameter to `BASIC` disables the collection of many of the important statistics required by Oracle Database features and functionality, including:

- Automatic Workload Repository (AWR) Snapshots
- Automatic Database Diagnostic Monitor (ADDM)
- All server-generated alerts
- Automatic SGA Memory Management
- Automatic optimizer statistics collection
- Object level statistics
- End to End Application Tracing (`V$CLIENT_STATS`)
- Database time distribution statistics (`V$SESS_TIME_MODEL` and `V$SYS_TIME_MODEL`)
- Service level statistics
- Buffer cache advisory
- MTTR advisory
- Shared pool sizing advisory
- Segment level statistics
- PGA Target advisory
- Timed statistics
- Monitoring of statistics

Note: Oracle strongly recommends that you do not disable these important features and functionality.

When the STATISTICS_LEVEL parameter is modified by ALTER SYSTEM, all advisories or statistics are dynamically turned on or off, depending on the new value of STATISTICS_LEVEL. When modified by ALTER SESSION, the following advisories or statistics are turned on or off in the local session only. Their system-wide state is not changed:

- Timed statistics
- Timed OS statistics
- Plan execution statistics

The V$STATISTICS_LEVEL view displays information about the status of the statistics or advisories controlled by the STATISTICS_LEVEL parameter. See "V$STATISTICS_LEVEL" on page 8-55.

STREAMS_POOL_SIZE

Property	Description
Parameter type	Big integer
Syntax	`STREAMS_POOL_SIZE = *integer* [K | M | G]`
Default value	0
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent

Oracle's Automatic Shared Memory Management feature manages the size of the Streams pool when the SGA_TARGET initialization parameter is set to a nonzero value. If the STREAMS_POOL_SIZE initialization parameter also is set to a nonzero value, then Automatic Shared Memory Management uses this value as a minimum for the Streams pool.

If the STREAMS_POOL_SIZE initialization parameter is set to a nonzero value, and the SGA_TARGET parameter is set to 0 (zero), then the Streams pool size is the value specified by the STREAMS_POOL_SIZE parameter, in bytes.

If both the STREAMS_POOL_SIZE and the SGA_TARGET initialization parameters are set to 0 (zero), then, by default, the first use of Streams in a database transfers an amount of memory equal to 10% of the shared pool from the buffer cache to the Streams pool.

See Also: *Oracle Streams Concepts and Administration*

TAPE_ASYNCH_IO

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Range of values	true | false

TAPE_ASYNCH_IO controls whether I/O to sequential devices (for example, backup or restore of Oracle data to or from tape) is asynchronous—that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans. If your platform supports asynchronous I/O to sequential devices, Oracle recommends that you leave this parameter set to its default. However, if the asynchronous I/O implementation is not stable, you can set TAPE_ASYNCH_IO to false to disable asynchronous I/O. If your platform does not support asynchronous I/O to sequential devices, this parameter has no effect.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on asynchronous I/O and on setting this parameter

THREAD

Property	Description
Parameter type	Integer
Default value	0
Modifiable	No
Range of values	0 to the maximum number of enabled threads
Real Application Clusters	If specified, multiple instances must have different values.

THREAD has been superseded by the INSTANCE_NAME and INSTANCE_NUMBER parameters, and will be made obsolete in a future release of the Oracle Database.

> **See Also:** "INSTANCE_NAME" on page 1-70 and "INSTANCE_NUMBER" on page 1-71

TIMED_OS_STATISTICS

Property	Description
Parameter type	Integer
Default value	If STATISTICS_LEVEL is set to ALL, then 5 If STATISTICS_LEVEL is set to BASIC or TYPICAL, then 0
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	Unlimited

TIMED_OS_STATISTICS specifies (in seconds) the interval at which Oracle collects operating system statistics when a request is made from the client to the server or when a request completes.

- On dedicated servers, Oracle collects operating system statistics at user logon and after each subsequent client invocation through the OCI into the Oracle server as a remote procedure call message.
- On shared servers, Oracle collects statistics when client calls to Oracle are processed.

A value of zero specifies that operating system statistics are not gathered. To collect statistics, set a value meaningful for your application and site needs.

> **Note:** Gathering operating system statistics is very expensive. Oracle recommends that you set this parameter in an ALTER SYSTEM statement rather than in the initialization parameter file, and that you reset the value to zero as soon as the needed statistics have been gathered.

TIMED_STATISTICS

Property	Description
Parameter type	Boolean
Default value	If STATISTICS_LEVEL is set to TYPICAL or ALL, then true If STATISTICS_LEVEL is set to BASIC, then false
Modifiable	ALTER SESSION, ALTER SYSTEM
Range of values	true | false

TIMED_STATISTICS specifies whether or not statistics related to time are collected.

Values:

- true

  The statistics are collected and stored in trace files or displayed in the V$SESSTATS and V$SYSSTATS dynamic performance views.

- false

  The value of all time-related statistics is set to zero. This setting lets Oracle avoid the overhead of requesting the time from the operating system. Normally, TIMED_STATISTICS should be false.

On some systems with very fast timer access, Oracle might enable timing even if this parameter is set to false. On these systems, setting the parameter to true can sometimes produce more accurate statistics for long-running operations.

See Also:

- *Oracle Database Performance Tuning Guide* for more information on setting this parameter and on performance diagnostic tools in general
- Appendix E, "Statistics Descriptions" indicates which statistics depend on the setting of this parameter.

TRACE_ENABLED

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	ALTER SYSTEM
Range of values	true | false

Property	Description
Real Application Clusters	The default value is TRUE. Oracle recommends that multiple instances have the same value.

TRACE_ENABLED controls tracing of the execution history, or code path, of Oracle. Oracle Support Services uses this information for debugging.

When TRACE_ENABLED is set to true, Oracle records information in specific files when errors occur.

Oracle records this information for all instances, even if only one instance terminates. This allows Oracle to retain diagnostics for an entire cluster.

Although the overhead incurred from this processing is not excessive, you can improve performance by setting TRACE_ENABLED to false. You might do this, for example, to meet high-end benchmark requirements. However, if you leave this parameter set to false, you may lose valuable diagnostic information. Therefore, always set TRACE_ENABLED to true to trace system problems and to reduce diagnostic efforts in the event of unexplained instance failures.

TRACEFILE_IDENTIFIER

Property	Description
Parameter type	String
Syntax	TRACEFILE_IDENTIFIER = "*traceid*"
Default value	There is no default value.
Modifiable	ALTER SESSION
Range of values	Any characters that can occur as part of a file name on the customer platform

TRACEFILE_IDENTIFIER specifies a custom identifier that becomes part of the Oracle Trace file name. Such a custom identifier is used to identify a trace file simply from its name and without having to open it or view its contents.

Each time this parameter is dynamically modified, the next trace dump will be written to a trace file which has the new parameter value embedded in its name. Trace file continuity information is automatically added to both the old and new trace files to indicate that these trace files belong to the same process.

This parameter can only be used to change the name of the foreground process' trace file; the background processes continue to have their trace files named in the regular format. For foreground processes, the TRACEID column of the V$PROCESS view contains the current value of the TRACEFILE_IDENTIFIER parameter. When this parameter value is set, the trace file name has the following format:

*sid*_ora_*pid*_*traceid*.trc

In this example, *sid* is the oracle instance ID, *pid* is the process ID, and *traceid* is the value of the TRACEFILE_IDENTIFIER parameter.

> **See Also:** This parameter is not supported on all operating systems. See your operating system-specific Oracle documentation for more information.

TRANSACTIONS

Property	Description
Parameter type	Integer
Default value	Derived: (1.1 * `SESSIONS`)
Modifiable	No
Range of values	4 to 2^{32}
Real Application Clusters	Multiple instances can have different values.

`TRANSACTIONS` specifies the maximum number of concurrent transactions. Greater values increase the size of the SGA and can increase the number of rollback segments allocated. The default value is greater than `SESSIONS` (and, in turn, `PROCESSES`) to allow for recursive transactions.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter
- *Oracle Real Application Clusters Administration and Deployment Guide* for the relationship of this parameter to the number of rollback segments

TRANSACTIONS_PER_ROLLBACK_SEGMENT

Property	Description
Parameter type	Integer
Default value	`5`
Modifiable	No
Range of values	1 to operating system-dependent
Real Application Clusters	Multiple instances can have different values.

`TRANSACTIONS_PER_ROLLBACK_SEGMENT` specifies the number of concurrent transactions you expect each rollback segment to have to handle. The minimum number of rollback segments acquired at startup is `TRANSACTIONS` divided by the value for this parameter. For example, if `TRANSACTIONS` is 101 and this parameter is 10, then the minimum number of rollback segments acquired would be the ratio 101/10, rounded up to 11.

You can acquire more rollback segments by naming them in the parameter `ROLLBACK_SEGMENTS`.

See Also:

- *Oracle Database Administrator's Guide* for information on setting this parameter.
- *Oracle Real Application Clusters Administration and Deployment Guide* for how Oracle acquires rollback segments.
- Your operating system-specific Oracle documentation for the range of values for this parameter.

UNDO_MANAGEMENT

Property	Description
Parameter type	String
Syntax	`UNDO_MANAGEMENT = { MANUAL | AUTO }`
Default value	`AUTO`
Modifiable	No
Basic	Yes
Real Application Clusters	Multiple instances must have the same value.

`UNDO_MANAGEMENT` specifies which undo space management mode the system should use. When set to `AUTO`, the instance starts in automatic undo management mode. In manual undo management mode, undo space is allocated externally as rollback segments.

UNDO_RETENTION

Property	Description
Parameter type	Integer
Default value	`900`
Modifiable	`ALTER SYSTEM`
Range of values	`0` to 2^{31} - 1
Real Application Clusters	Oracle recommends that multiple instances have the same value.

`UNDO_RETENTION` specifies (in seconds) the low threshold value of undo retention. For AUTOEXTEND undo tablespaces, the system retains undo for at least the time specified in this parameter, and automatically tunes the undo retention period to satisfy the undo requirements of the queries. For fixed- size undo tablespaces, the system automatically tunes for the maximum possible undo retention period, based on undo tablespace size and usage history, and ignores `UNDO_RETENTION` unless retention guarantee is enabled.

The setting of this parameter should account for any flashback requirements of the system. Automatic tuning of undo retention is not supported for LOBs. The RETENTION value for LOB columns is set to the value of the `UNDO_RETENTION` parameter.

The UNDO_RETENTION parameter can only be honored if the current undo tablespace has enough space. If an active transaction requires undo space and the undo tablespace does not have available space, then the system starts reusing unexpired undo space. This action can potentially cause some queries to fail with a "snapshot too old" message.

The amount of time for which undo is retained for the Oracle Database for the current undo tablespace can be obtained by querying the TUNED_UNDORETENTION column of the V$UNDOSTAT dynamic performance view.

See Also:

- *Oracle Database SQL Language Reference* for more information about creating undo tablespaces
- *Oracle Database Administrator's Guide* for more information about managing undo data

UNDO_TABLESPACE

Property	Description
Parameter type	String
Syntax	UNDO_TABLESPACE = *undoname*
Default value	The first available undo tablespace in the database.
Modifiable	ALTER SYSTEM
Range of values	Legal name of an existing undo tablespace
Basic	Yes
Real Application Clusters	Each instance must have a unique value for this parameter, when it is set.

UNDO_TABLESPACE specifies the undo tablespace to be used when an instance starts up. If this parameter is specified when the instance is in manual undo management mode, then an error will occur and startup will fail.

If the UNDO_TABLESPACE parameter is omitted, the first available undo tablespace in the database is chosen. If no undo tablespace is available, the instance will start without an undo tablespace. In such cases, user transactions will be executed using the SYSTEM rollback segment. You should avoid running in this mode under normal circumstances.

You can replace an undo tablespace with another undo tablespace while the instance is running.

See Also: *Oracle Database SQL Language Reference*.

USE_INDIRECT_DATA_BUFFERS

Property	Description
Parameter type	Boolean
Default value	false
Modifiable	No

Property	Description
Range of values	`true` | `false`

`USE_INDIRECT_DATA_BUFFERS` controls how the system global area (SGA) uses memory. It enables or disables the use of the extended buffer cache mechanism for 32-bit platforms that can support more than 4 GB of physical memory. On platforms that do not support this much physical memory, this parameter is ignored.

See Also:

- "LOCK_SGA" on page 1-79, "SHARED_MEMORY_ADDRESS" on page 1-147, and "HI_SHARED_MEMORY_ADDRESS" on page 1-68, which are other parameters that control how the SGA uses memory.
- *Oracle Database Concepts* for more information about the SGA.

USER_DUMP_DEST

Property	Description
Parameter type	String
Syntax	`USER_DUMP_DEST = {` *`pathname`* | *`directory`* `}`
Default value	Operating system-dependent
Modifiable	`ALTER SYSTEM`
Range of values	Any valid local path, directory, or disk

`USER_DUMP_DEST` specifies the pathname for a directory where the server will write debugging trace files on behalf of a user process.

For example, this directory might be set as follows:

- On MS-DOS: `C:\ORACLE\UTRC`
- On UNIX: `/oracle/utrc`
- On VMS: `DISK$UR3:[ORACLE.UTRC]`

Note: This parameter is ignored by the new diagnosability infrastructure introduced in Oracle Database 11g Release 1, which places trace and core files in a location controlled by the `DIAGNOSTIC_DEST` initialization parameter.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about the use of trace files
- Your operating system-specific Oracle documentation for the range of values

UTL_FILE_DIR

Property	Description
Parameter type	String
Syntax	`UTL_FILE_DIR = `*`pathname`*
Default value	There is no default value.
Modifiable	No
Range of values	Any valid directory path

`UTL_FILE_DIR` lets you specify one or more directories that Oracle should use for PL/SQL file I/O. If you are specifying multiple directories, you must repeat the `UTL_FILE_DIR` parameter for each directory on separate lines of the initialization parameter file.

All users can read or write to all files specified by this parameter. Therefore all PL/SQL users must be trusted with the information in the directories specified by this parameter.

Note: If you list multiple values, all entries of this parameter must be on contiguous lines of the parameter file. If you separate them with other parameters, Oracle will read only the last (contiguous) lines.

WORKAREA_SIZE_POLICY

Property	Description
Parameter type	String
Syntax	`WORKAREA_SIZE_POLICY = { AUTO | MANUAL }`
Default value	`AUTO`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`

`WORKAREA_SIZE_POLICY` specifies the policy for sizing work areas. This parameter controls the mode in which working areas are tuned.

Values:

- `AUTO`

  Work areas used by memory-intensive operators are sized automatically, based on the PGA memory used by the system, the target PGA memory set in `PGA_AGGREGATE_TARGET`, and the requirement of each individual operator.

- `MANUAL`

  The sizing of work areas is manual and based on the values of the `*_AREA_SIZE` parameter corresponding to the operation (for example, a sort uses `SORT_AREA_SIZE`). Specifying `MANUAL` may result in sub-optimal performance and poor PGA memory utilization.

XML_DB_EVENTS

Property	Description
Parameter type	String
Syntax	`XML_DB_EVENTS = { enable | disable }`
Default value	`enable`
Modifiable	`ALTER SESSION`, `ALTER SYSTEM`
Basic	No

`XML_DB_EVENTS` enables or disables XML DB Events.

Changing this parameter through an `ALTER SESSION` statement affects only the current session. Only users with the `XDBADMIN` role are allowed to change this parameter in a session.

Changing this parameter through an `ALTER SYSTEM` statement will make a system-wide change of this parameter; however, the change is only registered by new sessions started after the change. Currently running sessions need to be restarted to pick up the new change. Users must have the `ALTER SYSTEM` privilege to make a system-wide change.

Part II

Static Data Dictionary Views

This part describes data dictionary tables and views. These tables and views are called **static**, because they change only when a change is made to the data dictionary (for example, when a new table is created or when a user is granted new privileges). This part contains the following chapters:

- Chapter 2, "Static Data Dictionary Views: ALL_ALL_TABLES to ALL_OUTLINES"
- Chapter 3, "Static Data Dictionary Views: ALL_PART_COL_STATISTICS to DATABASE_PROPERTIES"
- Chapter 4, "Static Data Dictionary Views: DBA_2PC_NEIGHBORS to DBA_MVIEWS"
- Chapter 5, "Static Data Dictionary Views: DBA_NESTED_TABLE_COLS to USER_XML_VIEWS"

Note: Oracle also maintains tables that monitor ongoing database activity. These **dynamic performance tables** are described in Part III, "Dynamic Performance Views".

2

Static Data Dictionary Views: ALL_ALL_TABLES to ALL_OUTLINES

This chapter describes the first set (in alphabetical order) of static data dictionary views.The remaining static data dictionary views appear in alphabetical order in Chapter 3 through Chapter 5.

This chapter contains the following topics:

- About Static Data Dictionary Views
- Static Data Dictionary View Descriptions

About Static Data Dictionary Views

Data dictionary tables are not directly accessible, but you can access information in them through data dictionary views. To list the data dictionary views available to you, query the view DICTIONARY.

Many data dictionary tables have three corresponding views:

- An ALL_ view displays all the information accessible to the current user, including information from the current user's schema as well as information from objects in other schemas, if the current user has access to those objects by way of grants of privileges or roles.
- A DBA_ view displays all relevant information in the entire database. DBA_ views are intended only for administrators. They can be accessed only by users with the SELECT ANY TABLE privilege. This privilege is assigned to the DBA role when the system is initially installed.
- A USER_ view displays all the information from the schema of the current user. No special privileges are required to query these views.

The columns of the ALL_, DBA_, and USER_ views corresponding to a single data dictionary table are usually nearly identical. Therefore, these views are described in full only once in this chapter, at their first occurrence alphabetically, and are listed without full descriptions at their other occurrences.

Oracle Replication Views

A number of data dictionary views are relevant only if you are using Oracle Replication. Table 2–1 lists these views.

Table 2–1 Oracle Replication Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_REPCAT	DBA_REPCAT	USER_REPCAT
	DBA_REPCAT_EXCEPTIONS	
ALL_REPCAT_REFRESH_TEMPLATES	DBA_REPCAT_REFRESH_TEMPLATES	USER_REPCAT_REFRESH_TEMPLATES
ALL_REPCAT_TEMPLATE_OBJECTS	DBA_REPCAT_TEMPLATE_OBJECTS	USER_REPCAT_TEMPLATE_OBJECTS
ALL_REPCAT_TEMPLATE_PARMS	DBA_REPCAT_TEMPLATE_PARMS	USER_REPCAT_TEMPLATE_PARMS
ALL_REPCAT_TEMPLATE_SITES	DBA_REPCAT_TEMPLATE_SITES	USER_REPCAT_TEMPLATE_SITES
ALL_REPCAT_USER_AUTHORIZATIONS	DBA_REPCAT_USER_AUTHORIZATIONS	USER_REPCAT_USER_AUTHORIZATIONS
ALL_REPCAT_USER_PARM_VALUES	DBA_REPCAT_USER_PARM_VALUES	USER_REPCAT_USER_PARM_VALUES
ALL_REPCATLOG	DBA_REPCATLOG	USER_REPCATLOG
ALL_REPCOLUMN	DBA_REPCOLUMN	USER_REPCOLUMN
ALL_REPCOLUMN_GROUP	DBA_REPCOLUMN_GROUP	USER_REPCOLUMN_GROUP
ALL_REPCONFLICT	DBA_REPCONFLICT	USER_REPCONFLICT
ALL_REPDDL	DBA_REPDDL	USER_REPDDL
	DBA_REPEXTENSIONS	
ALL_REPFLAVOR_COLUMNS	DBA_REPFLAVOR_COLUMNS	USER_REPFLAVOR_COLUMNS
ALL_REPFLAVOR_OBJECTS	DBA_REPFLAVOR_OBJECTS	USER_REPFLAVOR_OBJECTS
ALL_REPFLAVORS	DBA_REPFLAVORS	USER_REPFLAVORS
ALL_REPGENERATED	DBA_REPGENERATED	USER_REPGENERATED
ALL_REPGENOBJECTS	DBA_REPGENOBJECTS	USER_REPGENOBJECTS
ALL_REPGROUP	DBA_REPGROUP	USER_REPGROUP
ALL_REPGROUP_PRIVILEGES	DBA_REPGROUP_PRIVILEGES	USER_REPGROUP_PRIVILEGES
ALL_REPGROUPED_COLUMN	DBA_REPGROUPED_COLUMN	USER_REPGROUPED_COLUMN
ALL_REPKEY_COLUMNS	DBA_REPKEY_COLUMNS	USER_REPKEY_COLUMNS
ALL_REPOBJECT	DBA_REPOBJECT	USER_REPOBJECT
ALL_REPPARAMETER_COLUMN	DBA_REPPARAMETER_COLUMN	USER_REPPARAMETER_COLUMN
ALL_REPPRIORITY	DBA_REPPRIORITY	USER_REPPRIORITY
ALL_REPPRIORITY_GROUP	DBA_REPPRIORITY_GROUP	USER_REPPRIORITY_GROUP
ALL_REPPROP	DBA_REPPROP	USER_REPPROP
ALL_REPRESOL_STATS_CONTROL	DBA_REPRESOL_STATS_CONTROL	USER_REPRESOL_STATS_CONTROL
ALL_REPRESOLUTION	DBA_REPRESOLUTION	USER_REPRESOLUTION
ALL_REPRESOLUTION_METHOD	DBA_REPRESOLUTION_METHOD	USER_REPRESOLUTION_METHOD
ALL_REPRESOLUTION_STATISTICS	DBA_REPRESOLUTION_STATISTICS	USER_REPRESOLUTION_STATISTICS

Table 2–1 (Cont.) Oracle Replication Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_REPSCHEMA	DBA_REPSCHEMA	USER_REPSCHEMA
ALL_REPSITES	DBA_REPSITES	USER_REPSITES
	DBA_REPSITES_NEW	

The following are additional Oracle Replication data dictionary views:

```
DEFCALL
DEFCALLDEST
DEFDEFAULTDEST
DEFERRCOUNT
DEFERROR
DEFLOB
DEFPROPAGATOR
DEFSCHEDULE
DEFTRAN
DEFTRANDEST
```

See Also: *Oracle Database Advanced Replication Management API Reference* for information about these views

Oracle Workspace Manager Views

A number of data dictionary views are relevant only if you are using Oracle Workspace Manager. Table 2–2 lists these views.

Table 2–2 Oracle Workspace Manager Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_MP_GRAPH_WORKSPACES		USER_MP_GRAPH_WORKSPACES
ALL_MP_PARENT_WORKSPACES		USER_MP_PARENT_WORKSPACES
ALL_REMOVED_WORKSPACES		USER_REMOVED_WORKSPACES
ALL_VERSION_HVIEW		
ALL_WM_CONS_COLUMNS		USER_WM_CONS_COLUMNS
ALL_WM_CONSTRAINTS		USER_WM_CONSTRAINTS
ALL_WM_IND_COLUMNS		USER_WM_IND_COLUMNS
ALL_WM_IND_EXPRESSIONS		USER_WM_IND_EXPRESSIONS
ALL_WM_LOCKED_TABLES		USER_WM_LOCKED_TABLES
ALL_WM_MODIFIED_TABLES		USER_WM_MODIFIED_TABLES
		USER_WM_PRIVS
ALL_WM_RIC_INFO		USER_WM_RIC_INFO
	DBA_WM_SYS_PRIVS	
ALL_WM_TAB_TRIGGERS		USER_WM_TAB_TRIGGERS
ALL_WM_VERSIONED_TABLES		USER_WM_VERSIONED_TABLES
ALL_WM_VT_ERRORS	DBA_WM_VT_ERRORS	USER_WM_VT_ERRORS
ALL_WORKSPACE_PRIVS		USER_WORKSPACE_PRIVS

Table 2–2 (Cont.) Oracle Workspace Manager Data Dictionary Views

ALL_ Views	DBA_ Views	USER_ Views
ALL_WORKSPACE_SAVEPOINTS		USER_WORKSPACE_SAVEPOINTS
	DBA_WORKSPACE_SESSIONS	
ALL_WORKSPACES	DBA_WORKSPACES	USER_WORKSPACES

The following are additional Oracle Workspace Manager data dictionary views:

```
ROLE_WM_PRIVS
WM_COMPRESS_BATCH_SIZES
WM_COMPRESSIBLE_TABLES
WM_EVENTS_INFO
WM_INSTALLATION
WM_REPLICATION_INFO
```

See Also: *Oracle Database Workspace Manager Developer's Guide* for information about these views

Recovery Catalog Views

The following data dictionary views are only available after you create an optional recovery catalog (which contains schemas containing information about backups) for use with Recovery Manager:

```
RC_ARCHIVED_LOG
RC_BACKUP_ARCHIVELOG_DETAILS
RC_BACKUP_ARCHIVELOG_SUMMARY
RC_BACKUP_CONTROLFILE
RC_BACKUP_CONTROLFILE_DETAILS
RC_BACKUP_CONTROLFILE_SUMMARY
RC_BACKUP_COPY_DETAILS
RC_BACKUP_COPY_SUMMARY
RC_BACKUP_CORRUPTION
RC_BACKUP_DATAFILE
RC_BACKUP_DATAFILE_DETAILS
RC_BACKUP_DATAFILE_SUMMARY
RC_BACKUP_FILES
RC_BACKUP_PIECE
RC_BACKUP_PIECE_DETAILS
RC_BACKUP_REDOLOG
RC_BACKUP_SET
RC_BACKUP_SET_DETAILS
RC_BACKUP_SET_SUMMARY
RC_BACKUP_SPFILE
RC_BACKUP_SPFILE_DETAILS
RC_BACKUP_SPFILE_SUMMARY
RC_CHECKPOINT
RC_CONTROLFILE_COPY
RC_COPY_CORRUPTION
RC_DATABASE
RC_DATABASE_BLOCK_CORRUPTION
RC_DATABASE_INCARNATION
RC_DATAFILE
RC_DATAFILE_COPY
```

```
RC_LOG_HISTORY
RC_OFFLINE_RANGE
RC_PROXY_ARCHIVEDLOG
RC_PROXY_ARCHIVELOG_DETAILS
RC_PROXY_ARCHIVELOG_SUMMARY
RC_PROXY_CONTROLFILE
RC_PROXY_COPY_DETAILS
RC_PROXY_COPY_SUMMARY
RC_PROXY_DATAFILE
RC_REDO_LOG
RC_REDO_THREAD
RC_RESTORE_POINT
RC_RESYNC
RC_RMAN_BACKUP_JOB_DETAILS
RC_RMAN_BACKUP_SUBJOB_DETAILS
RC_RMAN_BACKUP_TYPE
RC_RMAN_CONFIGURATION
RC_RMAN_OUTPUT
RC_RMAN_STATUS
RC_SITE
RC_STORED_SCRIPT
RC_STORED_SCRIPT_LINE
RC_TABLESPACE
RC_TEMPFILE
RC_UNUSABLE_BACKUPFILE_DETAILS
```

See Also: *Oracle Database Backup and Recovery Reference* for information about these views

Static Data Dictionary View Descriptions

The remainder of this chapter describes the static data dictionary views in alphabetical order.

ALL_ALL_TABLES

ALL_ALL_TABLES describes the object tables and relational tables accessible to the current user.

Related Views

- DBA_ALL_TABLES describes all object tables and relational tables in the database.
- USER_ALL_TABLES describes the object tables and relational tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs

Column	Datatype	NULL	Description
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
STATUS	VARCHAR2(8)		If a previous DROP TABLE operation failed, indicates whether the table is unusable (UNUSABLE) or valid (VALID)
PCT_FREE	NUMBER		Minimum percentage of free space in a block; NULL for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; NULL for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; NULL for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; NULL for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); NULL for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; NULL for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; NULL for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; NULL for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Indicates whether the table has been backed up since the last modification (Y) or not (N)
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of rows in the table that are chained from one data block to another or that have migrated to a new block, requiring a link to preserve the old rowid. This column is updated only after you analyze the table.
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)
TABLE_LOCK	VARCHAR2(8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the table
LAST_ANALYZED	DATE		Date on which the table was most recently analyzed

Column	Datatype	NULL	Description
PARTITIONED	VARCHAR2(3)		Indicates whether the table is partitioned (YES) or not (NO)
IOT_TYPE	VARCHAR2(12)		If the table is an index-organized table, indicates whether an overflow segment has been specified (IOT_OVERFLOW) or not (NULL)
OBJECT_ID_TYPE	VARCHAR2(16)		Indicates whether the object ID is USER-DEFINED or SYSTEM GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		If an object table, owner of the type from which the table is created
TABLE_TYPE	VARCHAR2(30)		If an object table, type of the table
TEMPORARY	VARCHAR2(1)		Indicates whether the table is temporary (Y) or not (N)
SECONDARY	VARCHAR2(1)		Indicates whether the table is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge to contain the contents of a domain index (Y) or not (N)
NESTED	VARCHAR2(3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2(7)		Buffer pool to be used for table blocks: ■ DEFAULT ■ KEEP ■ RECYCLE
ROW_MOVEMENT	VARCHAR2(8)		If a partitioned table, indicates whether row movement is enabled (ENABLED) or disabled (DISABLED)
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected by analyzing the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: SYS$SESSION - Rows are preserved for the duration of the session SYS$TRANSACTION - Rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2(8)		Indicates whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.skip_corrupt_blocks procedure.
MONITORING	VARCHAR2(3)		Indicates whether the table has the MONITORING attribute set (YES) or not (NO)
CLUSTER_OWNER	VARCHAR2(30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2(8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); NULL for partitioned tables
DROPPED	VARCHAR2(3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); NULL for partitioned tables

See Also:

- "DBA_ALL_TABLES" on page 4-30
- "USER_ALL_TABLES" on page 5-100

ALL_APPLY

ALL_APPLY displays information about the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY displays information about all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue from which the apply process dequeues
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue from which the apply process dequeues
APPLY_CAPTURED	VARCHAR2(3)		Indicates whether the apply process applies captured events (YES) or user-enqueued events (NO)
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set used by the apply process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set used by the apply process for filtering
APPLY_USER	VARCHAR2(30)		User who is applying events
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.
APPLY_TAG	RAW(2000)		Tag associated with redo log records that are generated when changes are made by the apply process
DDL_HANDLER	VARCHAR2(98)		Name of the user-specified DDL handler, which handles DDL logical change records
PRECOMMIT_HANDLER	VARCHAR2(98)		Name of the user-specified pre-commit handler
MESSAGE_HANDLER	VARCHAR2(98)		Name of the user-specified procedure that handles dequeued events other than logical change records
STATUS	VARCHAR2(8)		Status of the apply process: ■ DISABLED ■ ENABLED ■ ABORTED
MAX_APPLIED_MESSAGE_NUMBER	NUMBER		System change number (SCN) corresponding to the apply process high-watermark for the last time the apply process was stopped using the DBMS_APPLY_ADM.STOP_APPLY procedure with the force parameter set to false. The apply process high-watermark is the SCN beyond which no events have been applied.
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set used by the apply process for filtering
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set used by the apply process for filtering
STATUS_CHANGE_TIME	DATE		Time that the STATUS of the apply process was changed
ERROR_NUMBER	NUMBER		Error number if the apply process was aborted
ERROR_MESSAGE	VARCHAR2(4000)		Error message if the apply process was aborted

Column	Datatype	NULL	Description
MESSAGE_DELIVERY_MODE	VARCHAR2(10)		Reserved for future use

See Also: "DBA_APPLY" on page 4-31

ALL_APPLY_CONFLICT_COLUMNS

ALL_APPLY_CONFLICT_COLUMNS displays information about the conflict handlers on the tables accessible to the current user.

Related View

DBA_APPLY_CONFLICT_COLUMNS displays information about the conflict handlers on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the object on which the update conflict handler is defined
OBJECT_NAME	VARCHAR2(30)		Name of the object on which the update conflict handler is defined
METHOD_NAME	VARCHAR2(92)		Name of the update conflict handler used to resolve conflicts
RESOLUTION_COLUMN	VARCHAR2(4000)		Name of the column used to resolve conflicts
COLUMN_NAME	VARCHAR2(30)		Name of a column in the column list for the update conflict handler
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

See Also: "DBA_APPLY_CONFLICT_COLUMNS" on page 4-31

ALL_APPLY_DML_HANDLERS

ALL_APPLY_DML_HANDLERS displays information about the DML handlers on the tables accessible to the current user.

Related View

DBA_APPLY_DML_HANDLERS displays information about the DML handlers on all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object on which the DML handler is specified
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object on which the DML handler is specified
OPERATION_NAME	VARCHAR2(13)		Name of the DML operation for which the DML handler is used: ■ DEFAULT ■ INSERT ■ UPDATE ■ DELETE ■ LOB_UPDATE ■ ASSEMBLE_LOBS

Column	Datatype	NULL	Description
USER_PROCEDURE	VARCHAR2(98)		Name of the user-specified DML handler, which handles row logical change records that contain the DML operation in the OPERATION_NAME column on the object
ERROR_HANDLER	VARCHAR2(1)		Indicates whether the DML handler handles only the relevant row logical change records that result in apply errors (Y) or all relevant row logical change records (N)
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.
APPLY_NAME	VARCHAR2(30)		Name of the apply process for the given object
ASSEMBLE_LOBS	VARCHAR2(1)		Indicates whether LOB assembly is used for LOB columns in logical change records (LCRs) processed by the handler (Y) or not (N) LOB assembly combines multiple LCRs for a LOB column resulting from a single row change into one row LCR before passing the LCR to the handler.

See Also: "DBA_APPLY_DML_HANDLERS" on page 4-31

ALL_APPLY_ENQUEUE

ALL_APPLY_ENQUEUE displays information about the apply enqueue actions for the rules where the destination queue exists and is accessible to the current user.

Related View

DBA_APPLY_ENQUEUE displays information about the apply enqueue actions for all rules in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
DESTINATION_QUEUE_NAME	VARCHAR2(4000)		Name of the queue where events satisfying the rule will be enqueued

See Also: "DBA_APPLY_ENQUEUE" on page 4-31

ALL_APPLY_ERROR

ALL_APPLY_ERROR displays information about the error transactions generated by the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY_ERROR displays information about the error transactions generated by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)		Name of the apply process at the local database which processed the transaction
QUEUE_NAME	VARCHAR2(30)		Name of the queue at the local database from which the transaction was dequeued
QUEUE_OWNER	VARCHAR2(30)		Owner of the queue at the local database from which the transaction was dequeued

Column	Datatype	NULL	Description
LOCAL_TRANSACTION_ID	VARCHAR2(22)		Local transaction ID for the error transaction
SOURCE_DATABASE	VARCHAR2(128)		Database where the transaction originated
SOURCE_TRANSACTION_ID	VARCHAR2(22)		Original transaction ID at the source database
SOURCE_COMMIT_SCN	NUMBER		Original commit system change number (SCN) for the transaction at the source database
MESSAGE_NUMBER	NUMBER		Identifier for the event in the transaction that raised an error
ERROR_NUMBER	NUMBER		Error number of the error raised by the transaction
ERROR_MESSAGE	VARCHAR2(4000)		Error message of the error raised by the transaction
RECIPIENT_ID	NUMBER		User ID of the original user that applied the transaction
RECIPIENT_NAME	VARCHAR2(30)		Name of the original user that applied the transaction
MESSAGE_COUNT	NUMBER		Total number of events inside the error transaction
ERROR_CREATION_TIME	DATE		Time that the error was created

See Also: "DBA_APPLY_ERROR" on page 4-31

ALL_APPLY_EXECUTE

ALL_APPLY_EXECUTE displays information about the apply execute actions for the rules visible to the current user.

Related View

DBA_APPLY_EXECUTE displays information about the apply execute actions for all rules in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
EXECUTE_EVENT	VARCHAR2(2)		Indicates whether the event satisfying the rule is executed

See Also: "DBA_APPLY_EXECUTE" on page 4-31

ALL_APPLY_KEY_COLUMNS

ALL_APPLY_KEY_COLUMNS displays information about the substitute key columns for the tables accessible to the current user. Substitute key columns are set using the DBMS_APPLY_ADM.SET_KEY_COLUMNS procedure.

Related View

DBA_APPLY_KEY_COLUMNS displays information about the substitute key columns for all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object on which substitute key columns are set
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object on which substitute key columns are set

Column	Datatype	NULL	Description
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name of a column specified as a substitute key column
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

See Also: "DBA_APPLY_KEY_COLUMNS" on page 4-32

ALL_APPLY_PARAMETERS

ALL_APPLY_PARAMETERS displays information about the parameters for the apply processes that dequeue events from queues accessible to the current user.

Related View

DBA_APPLY_PARAMETERS displays information about the parameters for all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
PARAMETER	VARCHAR2(128)	NOT NULL	Name of the parameter
VALUE	VARCHAR2(4000)		Parameter value
SET_BY_USER	VARCHAR2(3)		Indicates whether the parameter value was set by the user (YES) or was not set by the user (NO). If NO for a parameter, then the parameter is set to its default value. If YES for a parameter, then the parameter may or may not be set to its default value.

See Also: "DBA_APPLY_PARAMETERS" on page 4-33

ALL_APPLY_PROGRESS

ALL_APPLY_PROGRESS displays information about the progress made by the apply processes that dequeue events from queues accessible to the current user. This view only contains information about captured events. It does not contain information about user-enqueued events.

Related View

DBA_APPLY_PROGRESS displays information about the progress made by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Global name of the source database of the changes that are applied by the apply process
APPLIED_MESSAGE_NUMBER	NUMBER	NOT NULL	Message number up to which all transactions have definitely been applied. This value is the low-watermark for the apply process. That is, messages with a commit message number less than or equal to this message number have definitely been applied, but some messages with a higher commit message number may also have been applied.
OLDEST_MESSAGE_NUMBER	NUMBER	NOT NULL	Earliest message number of the transactions currently being dequeued and applied

Column	Datatype	NULL	Description
APPLY_TIME	DATE		Time at which the message with the message number displayed in the APPLIED_MESSAGE_NUMBER column was applied
APPLIED_MESSAGE_CREATE_TIME	DATE		Time at which the message with the message number displayed in the APPLIED_MESSAGE_NUMBER column was created at its source database
OLDEST_TRANSACTION_ID	VARCHAR2(22)		Oldest transaction ID of interest. (useful for detecting long-running or large transactions)
SPILL_MESSAGE_NUMBER	NUMBER		Spill low-watermark. Any message with a lower SCN has either been applied or spilled to disk (it will be dequeued from the Streams queue and capture will not need to resend any logical change records (LCRs) with a lower SCN). Spilled messages may not have been applied yet.

See Also: "DBA_APPLY_PROGRESS" on page 4-33

ALL_APPLY_TABLE_COLUMNS

ALL_APPLY_TABLE_COLUMNS displays information about the destination table object columns for the tables accessible to the current user.

Related View

DBA_APPLY_TABLE_COLUMNS displays information about the destination table object columns for all tables in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table
OBJECT_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the column
COMPARE_OLD_ON_DELETE	VARCHAR2(3)		Indicates whether to Compare the old value of the column on deletes (YES) or not (NO)
COMPARE_OLD_ON_UPDATE	VARCHAR2(3)		Indicates whether to Compare the old value of the column on updates (YES) or not (NO)
APPLY_DATABASE_LINK	VARCHAR2(128)		For remote tables, name of the database link pointing to the remote database

See Also: "DBA_APPLY_TABLE_COLUMNS" on page 4-33

ALL_ARGUMENTS

ALL_ARGUMENTS lists the arguments of the procedures and functions that are accessible to the current user.

Related Views

- DBA_ARGUMENTS lists the arguments of the procedures and functions that are available in the database.
- USER_ARGUMENTS lists the arguments of the procedures and functions that are owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the procedure or function
PACKAGE_NAME	VARCHAR2(30)		Name of the package
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
OVERLOAD	VARCHAR2(40)		Indicates the *n*th overloading ordered by its appearance in the source; otherwise, it is NULL.
SUBPROGRAM_ID	NUMBER		Unique subprogram identifier
ARGUMENT_NAME	VARCHAR2(30)		If the argument is a scalar type, then the argument name is the name of the argument. A null argument name is used to denote a function return. If the function return or argument is a composite type, this view will have one row for each attribute of the composite type. Attributes are recursively expanded if they are composite. The meanings of ARGUMENT_NAME, POSITION, SEQUENCE, and DATA_LEVEL are interdependent. Together, as a tuple, they represent a node of a flattened tree. ARGUMENT_NAME can refer to any of the following: ■ Return type, if ARGUMENT_NAME is null and DATA_LEVEL = 0 ■ The argument that appears in the argument list if ARGUMENT_NAME is not null and DATA_LEVEL = 0 ■ Attribute name of the composite type if ARGUMENT_NAME is not null and DATA_LEVEL > 0; ■ A collection element type if ARGUMENT_NAME is null and DATA_LEVEL > 0;
POSITION	NUMBER	NOT NULL	If DATA_LEVEL is zero, then this column holds the position of this item in the argument list, or zero for a function return value. If DATA_LEVEL is greater than zero, then this column holds the position of this item with respect to its siblings at the same DATA_LEVEL. So, for a referenced record field, this is the index of the field within the record. For a referenced collection element, this is 1 (because collection elements do not have siblings.)
SEQUENCE	NUMBER	NOT NULL	Defines the sequential order of the argument and its attributes. Argument sequence starts from 1. Return type and its recursively expanded (preorder tree walk) attributes will come first, and each argument with its recursively expanded (preorder tree walk) attributes will follow.
DATA_LEVEL	NUMBER	NOT NULL	Nesting depth of the argument for composite types
DATA_TYPE	VARCHAR2(30)		Datatype of the argument
DEFAULTED	VARCHAR2(1)		Specifies whether or not the argument is defaulted
DEFAULT_VALUE	LONG		Reserved for future use
DEFAULT_LENGTH	NUMBER		Reserved for future use
IN_OUT	VARCHAR2(9)		Direction of the argument: ■ IN ■ OUT ■ IN/OUT
DATA_LENGTH	NUMBER		Length of the column (in bytes)
DATA_PRECISION	NUMBER		Length in decimal digits (NUMBER) or binary digits (FLOAT)
DATA_SCALE	NUMBER		Digits to the right of the decimal point in a number

Column	Datatype	NULL	Description
RADIX	NUMBER		Argument radix for a number
CHARACTER_SET_NAME	VARCHAR2(44)		Character set name for the argument
TYPE_OWNER	VARCHAR2(30)		Owner of the type of the argument
TYPE_NAME	VARCHAR2(30)		Name of the type of the argument. If the type is a package local type (that is, it is declared in a package specification), then this column displays the name of the package.
TYPE_SUBNAME	VARCHAR2(30)		Relevant only for package local types. Displays the name of the type declared in the package identified in the TYPE_NAME column.
TYPE_LINK	VARCHAR2(128)		Relevant only for package local types when the package identified in the TYPE_NAME column is a remote package. This column displays the database link used to refer to the remote package.
PLS_TYPE	VARCHAR2(30)		For numeric arguments, the name of the PL/SQL type of the argument. Null otherwise.
CHAR_LENGTH	NUMBER		Character limit for string datatypes
CHAR_USED	VARCHAR2(1)		Indicates whether the byte limit (B) or char limit (C) is official for the string

See Also:

- "DBA_ARGUMENTS" on page 4-34
- "USER_ARGUMENTS" on page 5-101

ALL_ASSEMBLIES

ALL_ASSEMBLIES provides information about assemblies accessible to the current user.

Related Views

- DBA_ASSEMBLIES provides information about all assemblies in the database.
- USER_ASSEMBLIES provides information about all assemblies owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the assembly
ASSEMBLY_NAME	VARCHAR2(30)	NOT NULL	Name of the assembly
FILE_SPEC	VARCHAR2(4000)		Operating system file specification of the assembly
SECURITY_LEVEL	VARCHAR2(10)		The maximum security level of the assembly
IDENTITY	VARCHAR2(4000)		The identity of the assembly
STATUS	VARCHAR2(7)		Status of the assembly

See Also:

- "DBA_ASSEMBLIES" on page 4-35
- "USER_ASSEMBLIES" on page 5-101

ALL_ASSOCIATIONS

ALL_ASSOCIATIONS describes user-defined statistics associated with objects accessible to the current user.

Related Views

- DBA_ASSOCIATIONS describes all user-defined statistics in the database.
- USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object for which the association is being defined
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object for which the association is being defined
COLUMN_NAME	VARCHAR2(30)		Column name in the object for which the association is being defined
OBJECT_TYPE	VARCHAR2(9)		Kind of object with which statistics are being associated: column, type, package or function, indextype, or domain index.
STATSTYPE_SCHEMA	VARCHAR2(30)		Owner of the statistics type
STATSTYPE_NAME	VARCHAR2(30)		Name of statistics type that contains the cost, selectivity or statistics functions
DEF_SELECTIVITY	NUMBER		Default selectivity of the object, if any
DEF_CPU_COST	NUMBER		Default CPU cost of the object, if any
DEF_IO_COST	NUMBER		Default I/O cost of the object, if any
DEF_NET_COST	NUMBER		Default networking cost of the object, if any
INTERFACE_VERSION	NUMBER		Identifies the version number of the ODCIStats interface. Value is 1 for statistics type implementing Oracle8*i* 8.1; 0 for types implementing Oracle9*i* 9.0.0.
MAINTENANCE_TYPE	VARCHAR2(14)		Specifies whether the object is system-managed or user-managed

See Also:

- "DBA_ASSOCIATIONS" on page 4-35
- "USER_ASSOCIATIONS" on page 5-101

ALL_AUDIT_POLICIES

ALL_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views accessible to the current user.

Related Views

- DBA_AUDIT_POLICIES describes all fine-grained auditing policies in the database.
- USER_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views owned by the current user. This view does not display the OBJECT_SCHEMA column.

Column	Datatype	NULL	Description
OBJECT_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the table or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the table or view
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
POLICY_TEXT	VARCHAR2(4000)		Audit condition
POLICY_COLUMN	VARCHAR2(30)		Relevant column
PF_SCHEMA	VARCHAR2(30)		Owner of the audit handler function
PF_PACKAGE	VARCHAR2(30)		Name of the package containing the audit handler function
PF_FUNCTION	VARCHAR2(30)		Name of the audit handler function
ENABLED	VARCHAR2(3)		Indicates whether the policy is enabled (YES) or disabled (NO)
SEL	VARCHAR2(3)		Indicates whether the policy is applied to queries on the object (YES) or not (NO)
INS	VARCHAR2(3)		Indicates whether the policy is applied to INSERT statements on the object (YES) or not (NO)
UPD	VARCHAR2(3)		Indicates whether the policy is applied to UPDATE statements on the object (YES) or not (NO)
DEL	VARCHAR2(3)		Indicates whether the policy is applied to DELETE statements on the object (YES) or not (NO)
AUDIT_TRAIL	VARCHAR2(11)		Indicates whether the SQL_TEXT and SQL_BIND columns in the fine-grained audit trail are populated for this policy (DB_EXTENDED) or not (DB)
POLICY_COLUMN_OPTION	VARCHAR2(11)		Indicates whether all columns in the AUDIT_COLUMN parameter (ALL_COLUMNS) or any of the columns in the AUDIT_COLUMN parameter (ANY_COLUMNS) are considered for triggering fine-grained auditing

See Also:

- "DBA_AUDIT_POLICIES" on page 4-38
- "USER_AUDIT_POLICIES" on page 5-101

ALL_AUDIT_POLICY_COLUMNS

ALL_AUDIT_POLICY_COLUMNS describes the fine-grained auditing policy columns on the tables and views accessible to the current user.

Related Views

- DBA_AUDIT_POLICY_COLUMNS describes all fine-grained auditing policy columns in the database.
- USER_AUDIT_POLICY_COLUMNS describes the fine-grained auditing policy columns on the tables and views owned by the current user.

Column	Datatype	NULL	Description
OBJECT_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the table or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the table or view
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
POLICY_COLUMN	VARCHAR2(30)	NOT NULL	Relevant column of the policy

See Also:

- "DBA_AUDIT_POLICY_COLUMNS" on page 4-38
- "USER_AUDIT_POLICY_COLUMNS" on page 5-101

ALL_AW_PS

ALL_AW_PS describes the page spaces in the analytic workspaces accessible to the current user.

Related Views

- DBA_AW_PS describes the page spaces in all analytic workspaces in the database.
- USER_AW_PS describes the page spaces in the analytic workspaces owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the analytic workspace
AW_NUMBER	NUMBER	NOT NULL	Number of the analytic workspace
AW_NAME	VARCHAR2(30)		Name of the analytic workspace
PSNUMBER	NUMBER(10)		Number of the page space
GENERATIONS	NUMBER		Number of active generations in the page space
MAXPAGES	NUMBER		Maximum pages allocated in the page space

See Also:

- "DBA_AW_PS" on page 4-50
- "USER_AW_PS" on page 5-102
- *Oracle OLAP User's Guide* for more information about these views

ALL_AWS

ALL_AWS describes the analytic workspaces accessible to the current user.

Related Views

- DBA_AWS describes all analytic workspaces in the database.
- USER_AWS describes the analytic workspaces owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the analytic workspace
AW_NUMBER	NUMBER		Number of the analytic workspace
AW_NAME	VARCHAR2(30)		Name of the analytic workspace
AW_VERSION	VARCHAR2(4)		Format version of the analytic workspace: ■ 9.1 ■ 10.1 ■ 10.2 ■ 11.1
PAGESPACES	NUMBER		Number of pagespaces in the analytic workspace

Column	Datatype	NULL	Description
GENERATIONS	NUMBER		Number of active generations in the analytic workspace
FROZEN	VARCHAR2(6)		Freeze state of the analytic workspace: - Frozen - NoThaw

See Also:

- "DBA_AWS" on page 4-50
- "USER_AWS" on page 5-102
- *Oracle OLAP User's Guide* for more information about these views

ALL_BASE_TABLE_MVIEWS

ALL_BASE_TABLE_MVIEWS describes the materialized views using materialized view logs accessible to the current user. A materialized view log can be created for a master, base table, or master materialized view. Query this view at the master site or the master materialized view site to show one row for each materialized view using a materialized view log.

Related Views

- DBA_BASE_TABLE_MVIEWS describes all materialized views using materialized view logs in the database.
- USER_BASE_TABLE_MVIEWS describes the materialized views using materialized view logs owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema in which the master table or the master materialized view was created
MASTER	VARCHAR2(30)		Name of the master table or the master materialized view
MVIEW_LAST_REFRESH_TIME	DATE		Date when the materialized view based on the master was last refreshed
MVIEW_ID	NUMBER		Unique identifier of the materialized view that is based on the master

See Also:

- "DBA_BASE_TABLE_MVIEWS" on page 4-50
- "USER_BASE_TABLE_MVIEWS" on page 5-102

ALL_CAPTURE

ALL_CAPTURE displays information about the capture processes that enqueue the captured changes into queues accessible to the current user.

Related View

DBA_CAPTURE displays information about all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue used for staging captured changes
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue used for staging captured changes
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set used by the capture process for filtering
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set
CAPTURE_USER	VARCHAR2(30)		Current user who is enqueuing captured messages
START_SCN	NUMBER		System change number (SCN) from which the capture process will start to capture changes
STATUS	VARCHAR2(8)		Status of the capture process: ▪ DISABLED ▪ ENABLED ▪ ABORTED
CAPTURED_SCN	NUMBER		System change number (SCN) of the last redo log record scanned
APPLIED_SCN	NUMBER		System change number (SCN) of the most recent message dequeued by the relevant apply processes. All changes below this SCN have been dequeued by all apply processes that apply changes captured by this capture process.
USE_DATABASE_LINK	VARCHAR2(3)		Indicates whether the source database name is used as the database link to connect to the source database from the downstream database (YES) or not (NO). If the capture process was created at the source database, then this column will be null.
FIRST_SCN	NUMBER		System change number (SCN) from which the capture process can be restarted
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database
SOURCE_DBID	NUMBER		Database ID of the source database
SOURCE_RESETLOGS_SCN	NUMBER		Resetlogs system change number (SCN) of the source database
SOURCE_RESETLOGS_TIME	NUMBER		Resetlogs time of the source database
LOGMINER_ID	NUMBER		Session ID of the LogMiner session associated with the capture process
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set used by the capture process for filtering
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set used by the capture process for filtering
MAX_CHECKPOINT_SCN	NUMBER		System change number (SCN) at which the last checkpoint was taken by the capture process
REQUIRED_CHECKPOINT_SCN	NUMBER		Lowest checkpoint SCN for which the capture process requires redo information
LOGFILE_ASSIGNMENT	VARCHAR2(8)		Logfile assignment type for the capture process: ▪ IMPLICIT ▪ EXPLICIT
STATUS_CHANGE_TIME	DATE		Time that the STATUS of the capture process was changed
ERROR_NUMBER	NUMBER		Error number if the capture process was aborted
ERROR_MESSAGE	VARCHAR2(4000)		Error message if the capture process was aborted
VERSION	VARCHAR2(64)		Version number of the capture process

Column	Datatype	NULL	Description
CAPTURE_TYPE	VARCHAR2(10)		Type of the capture process: ■ DOWNSTREAM ■ LOCAL
LAST_ENQUEUED_SCN	NUMBER		Last enqueued system change number (SCN)
CHECKPOINT_RETENTION_TIME	NUMBER		Checkpoint retention time **Note:** When the checkpoint retention time for a capture process is set to INFINITE, then the value displayed in this column is 4294967295.

See Also: "DBA_CAPTURE" on page 4-51

ALL_CAPTURE_EXTRA_ATTRIBUTES

ALL_CAPTURE_EXTRA_ATTRIBUTES displays information about the extra attributes for the capture processes accessible to the current user.

Related View

DBA_CAPTURE_EXTRA_ATTRIBUTES displays information about the extra attributes for all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
ATTRIBUTE_NAME	VARCHAR2(30)	NOT NULL	Name of the extra attribute
INCLUDE	VARCHAR2(3)		Indicates whether the extra attribute is included (YES) or not (NO)
ROW_ATTRIBUTE	VARCHAR2(3)		Indicates whether the extra attribute is a row LCR attribute (YES) or not (NO)
DDL_ATTRIBUTE	VARCHAR2(3)		Indicates whether the extra attribute is a DDL LCR attribute (YES) or not (NO)

See Also: "DBA_CAPTURE_EXTRA_ATTRIBUTES" on page 4-51

ALL_CAPTURE_PARAMETERS

ALL_CAPTURE_PARAMETERS displays information about the parameters for the capture processes that enqueue the captured changes into queues accessible to the current user.

Related View

DBA_CAPTURE_PARAMETERS displays information about the parameters for all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
PARAMETER	VARCHAR2(128)	NOT NULL	Name of the parameter
VALUE	VARCHAR2(4000)		Parameter value
SET_BY_USER	VARCHAR2(3)		Indicates whether the parameter value was set by the user (YES) or was not set by the user (NO). If NO, then the parameter is set to its default value. If YES, then the parameter may or may not be set to its default value.

See Also: "DBA_CAPTURE_PARAMETERS" on page 4-51

ALL_CAPTURE_PREPARED_DATABASE

`ALL_CAPTURE_PREPARED_DATABASE` displays information about when the local database was prepared for instantiation. If the local database was not prepared for instantiation, then this view contains no rows.

Related View

`DBA_CAPTURE_PREPARED_DATABASE` displays information about when the local database was prepared for instantiation.

Column	Datatype	NULL	Description
`TIMESTAMP`	`DATE`		Date and time at which the local database was ready to be instantiated
`SUPPLEMENTAL_LOG_DATA_PK`	`VARCHAR2(8)`		Status of database-level PRIMARY KEY COLUMNS supplemental logging: ■ `IMPLICIT` ■ `EXPLICIT` ■ `NO`
`SUPPLEMENTAL_LOG_DATA_UI`	`VARCHAR2(8)`		Status of database-level UNIQUE INDEX COLUMNS supplemental logging: ■ `IMPLICIT` ■ `EXPLICIT` ■ `NO`
`SUPPLEMENTAL_LOG_DATA_FK`	`VARCHAR2(8)`		Status of database-level FOREIGN KEY COLUMNS supplemental logging: ■ `IMPLICIT` ■ `EXPLICIT` ■ `NO`
`SUPPLEMENTAL_LOG_DATA_ALL`	`VARCHAR2(8)`		Status of database-level ALL COLUMNS supplemental logging: ■ `IMPLICIT` ■ `EXPLICIT` ■ `NO`

See Also:

- "DBA_CAPTURE_PREPARED_DATABASE" on page 4-51
- *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

ALL_CAPTURE_PREPARED_SCHEMAS

`ALL_CAPTURE_PREPARED_SCHEMAS` displays information about the schemas prepared for instantiation that are accessible to the current user at the local database.

Related View

`DBA_CAPTURE_PREPARED_SCHEMAS` displays information about all schemas prepared for instantiation at the local database.

Column	Datatype	NULL	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Name of the schema that is ready to be instantiated
TIMESTAMP	DATE		Date and time at which the schema was ready to be instantiated
SUPPLEMENTAL_LOG_DATA_PK	VARCHAR2(8)		Status of schema-level PRIMARY KEY COLUMNS supplemental logging: ■ IMPLICIT ■ EXPLICIT ■ NO
SUPPLEMENTAL_LOG_DATA_UI	VARCHAR2(8)		Status of schema-level UNIQUE INDEX COLUMNS supplemental logging: ■ IMPLICIT ■ EXPLICIT ■ NO
SUPPLEMENTAL_LOG_DATA_FK	VARCHAR2(8)		Status of schema-level FOREIGN KEY COLUMNS supplemental logging: ■ IMPLICIT ■ EXPLICIT ■ NO
SUPPLEMENTAL_LOG_DATA_ALL	VARCHAR2(8)		Status of schema-level ALL COLUMNS supplemental logging: ■ IMPLICIT ■ EXPLICIT ■ NO

See Also:

- "DBA_CAPTURE_PREPARED_SCHEMAS" on page 4-51
- *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

ALL_CAPTURE_PREPARED_TABLES

ALL_CAPTURE_PREPARED_TABLES displays information about the tables prepared for instantiation that are accessible to the current user at the local database.

Related View

DBA_CAPTURE_PREPARED_TABLES displays information about all tables prepared for instantiation at the local database.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table that is ready to be instantiated
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table that is ready to be instantiated
SCN	NUMBER	NOT NULL	Smallest system change number (SCN) for which the table can be instantiated
TIMESTAMP	DATE		Date and time at which the table was ready to be instantiated

Column	Datatype	NULL	Description
SUPPLEMENTAL_LOG_DATA_PK	VARCHAR2(8)		Status of table-level PRIMARY KEY COLUMNS supplemental logging: ▪ IMPLICIT ▪ EXPLICIT ▪ NO
SUPPLEMENTAL_LOG_DATA_UI	VARCHAR2(8)		Status of table-level UNIQUE INDEX COLUMNS supplemental logging: ▪ IMPLICIT ▪ EXPLICIT ▪ NO
SUPPLEMENTAL_LOG_DATA_FK	VARCHAR2(8)		Status of table-level FOREIGN KEY COLUMNS supplemental logging: ▪ IMPLICIT ▪ EXPLICIT ▪ NO
SUPPLEMENTAL_LOG_DATA_ALL	VARCHAR2(8)		Status of table-level ALL COLUMNS supplemental logging: ▪ IMPLICIT ▪ EXPLICIT ▪ NO

See Also:

- "DBA_CAPTURE_PREPARED_TABLES" on page 4-51
- *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

ALL_CATALOG

ALL_CATALOG lists all tables, clusters, views, synonyms, and sequences accessible to the current user.

Related Views

- DBA_CATALOG lists all tables, clusters, views, synonyms, and sequences in the entire database.
- USER_CATALOG lists all tables, clusters, views, synonyms, and sequences in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED
TABLE_TYPE	VARCHAR2(11)		Type of the TABLE, CLUSTER, VIEW, SYNONYM, SEQUENCE, or UNDEFINED

ALL_CHANGE_PROPAGATION_SETS

ALL_CHANGE_PROPAGATION_SETS displays information about the Change Data Capture propagated change sets.

Column	Datatype	NULL	Description
PROPAGATION_SOURCE_NAME	VARCHAR2(30)	NOT NULL	Name of the change source
PROPAGATION_NAME	VARCHAR2(30)	NOT NULL	Name of the propagation
STAGING_DATABASE	VARCHAR2(128)	NOT NULL	Name of the staging database for the propagation
CHANGE_SET_PUBLISHER	VARCHAR2(30)	NOT NULL	Publisher of the distributed change set
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the distributed change set

ALL_CHANGE_PROPAGATIONS

ALL_CHANGE_PROPAGATIONS displays information about Change Data Capture propagations.

Column	Datatype	NULL	Description
PROPAGATION_SOURCE_NAME	VARCHAR2(30)	NOT NULL	Name of the change source
PROPAGATION_NAME	VARCHAR2(30)	NOT NULL	Name of the propagation
STAGING_DATABASE	VARCHAR2(128)	NOT NULL	Name of the staging database for the propagation
DESTINATION_QUEUE_PUBLISHER	VARCHAR2(30)	NOT NULL	Owner of the destination queue
DESTINATION_QUEUE	VARCHAR2(30)	NOT NULL	Name of the destination queue

ALL_CHANGE_SETS

ALL_CHANGE_SETS displays information about Change Data Capture change sets.

Column	Datatype	NULL	Description
SET_NAME	VARCHAR2(30)	NOT NULL	Name of the change set
CHANGE_SOURCE_NAME	VARCHAR2(30)	NOT NULL	Change source to which the change set belongs
BEGIN_DATE	DATE		Starting point for capturing change data
END_DATE	DATE		Stopping point for capturing change data
BEGIN_SCN	NUMBER		Reserved for future use
END_SCN	NUMBER		Reserved for future use
FRESHNESS_DATE	DATE		Reserved for future use
FRESHNESS_SCN	NUMBER		Reserved for future use
ADVANCE_ENABLED	VARCHAR2(1)		Indicates whether the change set is enabled (Y) or disabled (N), or whether capture is disabled and apply is enabled (C)
IGNORE_DDL	VARCHAR2(1)		Reserved for future use
CREATED	DATE	NOT NULL	Creation date of the change set
ROLLBACK_SEGMENT_NAME	VARCHAR2(30)		Reserved for future use
ADVANCING	VARCHAR2(1)	NOT NULL	Reserved for future use
PURGING	VARCHAR2(1)	NOT NULL	Reserved for future use
LOWEST_SCN	NUMBER		Low-watermark for change data in the change set
TABLESPACE	VARCHAR2(30)		Reserved for future use
CAPTURE_ENABLED	VARCHAR2(1)	NOT NULL	Indicates whether capture is enabled for the change set (Y) or not (N)

Column	Datatype	NULL	Description
STOP_ON_DDL	VARCHAR2(1)	NOT NULL	Indicates whether the change set stops on DDL (Y) or not (N)
CAPTURE_ERROR	VARCHAR2(1)	NOT NULL	Indicates whether there is a capture error (Y) or not (N)
CAPTURE_NAME	VARCHAR2(30)		Name of the Streams capture process
QUEUE_NAME	VARCHAR2(30)		Name of the Streams queue
QUEUE_TABLE_NAME	VARCHAR2(30)		Name of the Streams queue table
APPLY_NAME	VARCHAR2(30)		Name of the Streams apply process
SET_DESCRIPTION	VARCHAR2(255)		Description of the change set
PUBLISHER	VARCHAR2(30)		Publisher of the change set
LOWEST_TIMESTAMP	DATE		Timestamp of the record with the lowest SCN in the change set
MAP_NAME	VARCHAR2(30)		Name of the mapping table; maps SCN to timestamp

ALL_CHANGE_SOURCES

ALL_CHANGE_SOURCES displays information about Change Data Capture change sources.

Column	Datatype	NULL	Description
SOURCE_NAME	VARCHAR2(30)	NOT NULL	Name of the change source
DBID	NUMBER		Database identifier
LOG_DIRECTORY	VARCHAR2(2000)		Log file directory location
LOGFILE_PATTERN	VARCHAR2(30)		Wildcard pattern for log file names
SOURCE_DESCRIPTION	VARCHAR2(255)		Description of the change source
CREATED	DATE	NOT NULL	Creation date of the change source
SOURCE_TYPE	VARCHAR2(18)		Capture mode of the change source: ■ AUTOLOG ■ HOTLOG ■ SYNCHRONOUS ■ DISTRIBUTED HOTLOG ■ AUTOLOG ONLINE
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database
FIRST_SCN	NUMBER		SCN of a LogMiner dictionary at which capture can begin
PUBLISHER	VARCHAR2(30)		Publisher of the change source
CAPTURE_NAME	VARCHAR2(30)		Name of the Streams capture process
CAPTURE_QUEUE_NAME	VARCHAR2(30)		Name of the Streams capture queue name
CAPTURE_QUEUE_TABLE_NAME	VARCHAR2(30)		Name of the Streams capture table
SOURCE_ENABLED	VARCHAR2(1)		Indicates whether the change source is enabled (Y) or not (N)

ALL_CHANGE_TABLES

ALL_CHANGE_TABLES displays information about Change Data Capture change tables.

Column	Datatype	NULL	Description
CHANGE_TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the change table
CHANGE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the change table
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the change set to which the change table belongs
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Owner of the source table for the change table
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the source table for the change table
CREATED	DATE	NOT NULL	Creation date of the change table
CREATED_SCN	NUMBER		Creation SCN of the change table
CAPTURED_VALUES	VARCHAR2(1)	NOT NULL	Indicates whether old (O), new (N), or both (B) update values are captured
PUB_ID	NUMBER	NOT NULL	Publication ID displayed to users for the change table

ALL_CLUSTER_HASH_EXPRESSIONS

ALL_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters accessible to the current user.

Related Views

- DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database.
- USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
HASH_EXPRESSION	LONG		Text of hash function of the hash cluster

ALL_CLUSTERS

ALL_CLUSTERS describes all clusters accessible to the current user.

Related Views

- DBA_CLUSTERS describes all clusters in the database.
- USER_CLUSTERS describes all clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Name of the cluster
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the cluster
PCT_FREE	NUMBER		Minimum percentage of free space in a block
PCT_USED	NUMBER		Minimum percentage of used space in a block
KEY_SIZE	NUMBER		Estimated size of cluster key plus associated rows
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions

Column	Datatype	NULL	Description
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
AVG_BLOCKS_PER_KEY	NUMBER		Number of blocks in the table divided by number of cluster keys
CLUSTER_TYPE	VARCHAR2(5)		Type of cluster: B*-Tree index or hash
FUNCTION	VARCHAR2(15)		If a hash cluster, the hash function
HASHKEYS	NUMBER		If a hash cluster, the number of hash keys (hash buckets)
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the cluster
INSTANCES	VARCHAR2(10)		Number of instances across which the cluster is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the cluster is to be cached in the buffer cache (Y) or not (N)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the cluster
SINGLE_TABLE	VARCHAR2(5)		Indicates whether this is a single-table cluster (Y) or not (N)
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)

ALL_COL_COMMENTS

ALL_COL_COMMENTS displays comments on the columns of the tables and views accessible to the current user.

Related Views

- DBA_COL_COMMENTS displays comments on the columns of all tables and views in the database.
- USER_COL_COMMENTS displays comments on the columns of the tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
COMMENTS	VARCHAR2(4000)		Comment on the column

See Also:

- "DBA_COL_COMMENTS" on page 4-53
- "USER_COL_COMMENTS" on page 5-103

ALL_COL_PENDING_STATS

ALL_COL_PENDING_STATS describes the pending statistics of the columns accessible to the current user.

Related Views

- DBA_COL_PENDING_STATS describes the pending statistics of all columns in the database.
- USER_COL_PENDING_STATS describes the pending statistics of the columns owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
COLUMN_NAME	VARCHAR2(30)		Name of the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of NULLs in the column
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the column
LAST_ANALYZED	DATE		Most recent date on which the column was analyzed

See Also:

- "DBA_COL_PENDING_STATS" on page 4-53
- "USER_COL_PENDING_STATS" on page 5-103

ALL_COL_PRIVS

ALL_COL_PRIVS describes the following types of grants:

- Column object grants for which the current user is the object owner, grantor, or grantee
- Column object grants for which an enabled role or PUBLIC is the grantee

Related Views

- DBA_COL_PRIVS describes all column object grants in the database.
- USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also:

- "DBA_COL_PRIVS" on page 4-53
- "USER_COL_PRIVS" on page 5-103

ALL_COL_PRIVS_MADE

ALL_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner or grantor.

Related View

USER_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: "USER_COL_PRIVS_MADE" on page 5-103

ALL_COL_PRIVS_RECD

ALL_COL_PRIVS_RECD describes the following types of grants:

- Column object grants for which the current user is the grantee
- Column object grants for which an enabled role or PUBLIC is the grantee

Related View

USER_COL_PRIVS_RECD describes the column object grants for which the current user is the grantee. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NUL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: "USER_COL_PRIVS_RECD" on page 5-103

ALL_COLL_TYPES

ALL_COLL_TYPES describes all named collection types (varrays and nested tables) accessible to the current user.

Related Views

- DBA_COLL_TYPES describes all named collection types in the database.
- USER_COLL_TYPES describes all named collection types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the collection
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the collection
COLL_TYPE	VARCHAR2(30)	NOT NULL	Description of the collection, such as VARYING ARRAY, [nested] TABLE
UPPER_BOUND	NUMBER		For varrays only, maximum size
ELEM_TYPE_MOD	VARCHAR2(7)		Type modifier of the element
ELEM_TYPE_OWNER	VARCHAR2(30)		Owner of the type upon which the collection is based. This value is useful primarily in the case of a user-defined type.
ELEM_TYPE_NAME	VARCHAR2(30)		Name of the datatype or user-defined type upon which the collection is based
LENGTH	NUMBER		Length of CHAR elements or maximum length of VARCHAR or VARCHAR2 elements
PRECISION	NUMBER		Decimal precision of NUMBER or DECIMAL elements; binary precision of FLOAT elements
SCALE	NUMBER		Scale of NUMBER or DECIMAL elements
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set (CHAR_CS | NCHAR_CS)
ELEM_STORAGE	VARCHAR2(7)		Obsolete column
NULLS_STORED	VARCHAR2(3)		Obsolete column

ALL_CONS_COLUMNS

ALL_CONS_COLUMNS describes columns that are accessible to the current user and that are specified in constraints.

Related Views

- DBA_CONS_COLUMNS describes all columns in the database that are specified in constraints.
- USER_CONS_COLUMNS describes columns that are owned by the current user and that are specified in constraints.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the constraint definition
CONSTRAINT_NAME	VARCHAR2(30)	NOT NULL	Name of the constraint definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table with constraint definition
COLUMN_NAME	VARCHAR2(4000)		Name of the column or attribute of the object type column specified in the constraint definition **Note:** If you create a constraint on a user-defined REF column, the system creates the constraint on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: "REF_name"."attribute"
POSITION	NUMBER		Original position of column or attribute in the definition of the object

ALL_CONS_OBJ_COLUMNS

ALL_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables accessible to the current user.

Related Views

- DBA_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in all tables in the database.
- USER_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table containing the object column or attribute
COLUMN_NAME	VARCHAR2(4000)		Fully qualified name of the object column or attribute
CONS_TYPE_OWNER	VARCHAR2(30)		Owner of the type that the column (or element) is constrained to
CONS_TYPE_NAME	VARCHAR2(30)		Name of the type that the column (or element) is constrained to
CONS_TYPE_ONLY	VARCHAR2(15)		Indicates whether the column (or element) is constrained to ONLY type (Y) or not (N)

See Also:

- "DBA_CONS_OBJ_COLUMNS" on page 4-58
- "USER_CONS_OBJ_COLUMNS" on page 5-104

ALL_CONSTRAINTS

ALL_CONSTRAINTS describes constraint definitions on tables accessible to the current user.

Related Views

- `DBA_CONSTRAINTS` describes all constraint definitions in the database.
- `USER_CONSTRAINTS` describes constraint definitions on tables in the current user's schema.

Column	Datatype	NULL	Description
`OWNER`	`VARCHAR2(30)`	`NOT NULL`	Owner of the constraint definition
`CONSTRAINT_NAME`	`VARCHAR2(30)`	`NOT NULL`	Name of the constraint definition
`CONSTRAINT_TYPE`	`VARCHAR2(1)`		Type of constraint definition: ■ `C` (check constraint on a table) ■ `P` (primary key) ■ `U` (unique key) ■ `R` (referential integrity) ■ `V` (with check option, on a view) ■ `O` (with read only, on a view)
`TABLE_NAME`	`VARCHAR2(30)`	`NOT NULL`	Name associated with the table (or view) with constraint definition
`SEARCH_CONDITION`	`LONG`		Text of search condition for a check constraint
`R_OWNER`	`VARCHAR2(30)`		Owner of table referred to in a referential constraint
`R_CONSTRAINT_NAME`	`VARCHAR2(30)`		Name of the unique constraint definition for referenced table
`DELETE_RULE`	`VARCHAR2(9)`		Delete rule for a referential constraint (`CASCADE` or `NO ACTION`)
`STATUS`	`VARCHAR2(8)`		Enforcement status of constraint (`ENABLED` or `DISABLED`)
`DEFERRABLE`	`VARCHAR2(14)`		Whether the constraint is deferrable
`DEFERRED`	`VARCHAR2(9)`		Whether the constraint was initially deferred
`VALIDATED`	`VARCHAR2(13)`		Whether all data obeys the constraint (`VALIDATED` or `NOT VALIDATED`)
`GENERATED`	`VARCHAR2(14)`		Whether the name of the constraint is user or system generated
`BAD`	`VARCHAR2(3)`		A `YES` value indicates that this constraint specifies a century in an ambiguous manner. To avoid errors resulting from this ambiguity, rewrite the constraint using the `TO_DATE` function with a four-digit year. **See Also:** the `TO_DATE` function in *Oracle Database SQL Language Reference* and *Oracle Database Advanced Application Developer's Guide*
`RELY`	`VARCHAR2(4)`		Whether an enabled constraint is enforced or unenforced. **See Also:** the *constraints* in *Oracle Database SQL Language Reference*
`LAST_CHANGE`	`DATE`		When the constraint was last enabled or disabled
`INDEX_OWNER`	`VARCHAR2(30)`		Name of the user owning the index
`INDEX_NAME`	`VARCHAR2(30)`		Name of the index (only shown for unique and primary-key constraints)
`INVALID`	`VARCHAR2(7)`		Whether the constraint is invalid
`VIEW_RELATED`	`VARCHAR2(14)`		Whether the constraint depends on a view

ALL_CONTEXT

ALL_CONTEXT describes all context namespaces in the current session for which attributes and values have been specified using the DBMS_SESSION.SET_CONTEXT procedure. This view does not display the TYPE column.

Related View

DBA_CONTEXT describes all context namespaces defined in the database, regardless whether any attributes have been specified for them using the DBMS_SESSION.SET_CONTEXT procedure.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	Name of the context namespace
SCHEMA	VARCHAR2(30)	NOT NULL	Schema name of the designated package that can set attributes using this namespace
PACKAGE	VARCHAR2(30)	NOT NULL	Package name of the designated package that can set attributes using this namespace
TYPE	VARCHAR2(22)		Type of the context: ■ ACCESSED LOCALLY ■ INITIALIZED EXTERNALLY ■ ACCESSED GLOBALLY ■ INITIALIZED GLOBALLY

ALL_CUBE_ATTR_VISIBILITY

ALL_CUBE_ATTR_VISIBILITY describes the OLAP attributes visible for the dimensions, hierarchies, and levels accessible to the current user.

Related Views

- DBA_CUBE_ATTR_VISIBILITY describes all OLAP attributes visible for the dimensions, hierarchies, and levels in the database.
- USER_CUBE_ATTR_VISIBILITY describes the OLAP attributes visible for the dimensions, hierarchies, and levels owned by the current user. This view does not display the OWNER column.

See Also:

- "DBA_CUBE_ATTR_VISIBILITY" on page 4-60
- "USER_CUBE_ATTR_VISIBILITY" on page 5-105

ALL_CUBE_ATTRIBUTES

ALL_CUBE_ATTRIBUTES describes the attributes for the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_ATTRIBUTES describes the attributes for all OLAP cube dimensions in the database.
- USER_CUBE_ATTRIBUTES describes the attributes for the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension (such as TIME)
ATTRIBUTE_NAME	VARCHAR2(30)	NOT NULL	Name of an attribute of the dimension (such as LONG_DESCRIPTION or END_DATE)
TARGET_DIMENSION_NAME	VARCHAR2(30)		Name of the target dimension of the attribute
ATTRIBUTE_ROLE	VARCHAR2(17)		Special role this attribute plays; NULL if none: ■ SHORT_DESCRIPTION ■ LONG_DESCRIPTION ■ DESCRIPTION ■ TIME_SPAN ■ END_DATE
DESCRIPTION	NVARCHAR2(300)		Description of the attribute in the session language
DATA_TYPE	VARCHAR2(106)		Data type of the attribute, (such as VARCHAR2 or FLOAT)
DATA_LENGTH	NUMBER	NOT NULL	Length of a text data type
DATA_PRECISION	NUMBER		Precision of a numeric data type
DATA_SCALE	NUMBER		Scale of a numeric data type

See Also:

- "DBA_CUBE_ATTRIBUTES" on page 4-60
- "USER_CUBE_ATTRIBUTES" on page 5-105

ALL_CUBE_BUILD_PROCESSES

ALL_CUBE_BUILD_PROCESSES describes the OLAP build processes and maintenance scripts accessible to the current user.

Related Views

- DBA_CUBE_BUILD_PROCESSES describes all OLAP build processes and maintenance scripts in the database.
- USER_CUBE_BUILD_PROCESSES describes the OLAP build processes and maintenance scripts owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the build process
BUILD_PROCESS_NAME	VARCHAR2(30)	NOT NULL	Name of the build process
BUILD_PROCESS	CLOB		Syntax of the build process
DESCRIPTION	NVARCHAR2(300)		Description of the build process in the session language

See Also:

- "DBA_CUBE_BUILD_PROCESSES" on page 4-60
- "USER_CUBE_BUILD_PROCESSES" on page 5-105

ALL_CUBE_CALCULATED_MEMBERS

ALL_CUBE_CALCULATED_MEMBERS describes the calculated members for the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_CALCULATED_MEMBERS describes the calculated members for all OLAP cube dimensions in the database.
- USER_CUBE_CALCULATED_MEMBERS describes the calculated members for the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension
MEMBER_NAME	VARCHAR2(30)	NOT NULL	Name of a calculated member in the cube dimension
IS_CUSTOM_AGGREGATE	VARCHAR2(3)		Indicates whether the calculated member is a custom aggregate (YES) or not (NO)
STORAGE_TYPE	VARCHAR2(10)		Storage type of the calculated member: ■ DYNAMIC - Value of the member is calculated for a query ■ PRECOMPUTE - Value of the member is calculated and stored during data maintenance
EXPRESSION	CLOB		Expression used to generate the value of the calculated member

See Also:

- "DBA_CUBE_CALCULATED_MEMBERS" on page 4-60
- "USER_CUBE_CALCULATED_MEMBERS" on page 5-105

ALL_CUBE_DIM_LEVELS

ALL_CUBE_DIM_LEVELS describes the OLAP dimension levels accessible to the current user.

Related Views

- DBA_CUBE_DIM_LEVELS describes all OLAP dimension levels in the database.
- USER_CUBE_DIM_LEVELS describes the OLAP dimension levels owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a dimension, such as CUSTOMER
LEVEL_NAME	VARCHAR2(30)	NOT NULL	Name of a level in the dimension, such as WAREHOUSE
DESCRIPTION	NVARCHAR2(300)		Description of the dimension level in the session language

See Also:

- "DBA_CUBE_DIM_LEVELS" on page 4-60
- "USER_CUBE_DIM_LEVELS" on page 5-105

ALL_CUBE_DIM_MODELS

ALL_CUBE_DIM_MODELS describes the models for the OLAP dimensions accessible to the current user.

Related Views

- DBA_CUBE_DIM_MODELS describes the models for all OLAP dimensions in the database.
- USER_CUBE_DIM_MODELS describes the models for the OLAP dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension
MODEL_NAME	VARCHAR2(30)	NOT NULL	Name of a model for the cube dimension
DESCRIPTION	NVARCHAR2(300)		Description of the model in the session language

See Also:

- "DBA_CUBE_DIM_MODELS" on page 4-60
- "USER_CUBE_DIM_MODELS" on page 5-105

ALL_CUBE_DIM_VIEW_COLUMNS

ALL_CUBE_DIM_VIEW_COLUMNS describes the columns of the relational views of the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_DIM_VIEW_COLUMNS describes the columns of the relational views of all OLAP cube dimensions in the database.
- USER_CUBE_DIM_VIEW_COLUMNS describes the columns of the relational views of the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)		Name of a cube dimension, such as PRODUCT
VIEW_NAME	VARCHAR2(30)		Name of a view of the dimension, such as PRODUCT_VIEW
COLUMN_NAME	VARCHAR2(30)		Name of a column in the view, such as LONG_DESCRIPTION or WAREHOUSE_ID

Column	Datatype	NULL	Description
COLUMN_TYPE	VARCHAR2(10)		Type of the column: ■ KEY - A key of the dimension view (that is, the dimension value itself) ■ LEVEL_NAME - Name of the level (if any) corresponding to a row in the view ■ DIM_ORDER - A column by which the results may be ordered (if present) ■ ATTRIBUTE - An attribute owned by the dimension
OBJECT_NAME	VARCHAR2(30)		Name of the level or attribute represented in the column, such as LONG_DESCRIPTION or WAREHOUSE_ID

See Also:

- "DBA_CUBE_DIM_VIEW_COLUMNS" on page 4-60
- "USER_CUBE_DIM_VIEW_COLUMNS" on page 5-106

ALL_CUBE_DIM_VIEWS

ALL_CUBE_DIM_VIEWS describes the relational views of the OLAP dimensions accessible to the current user.

Related Views

- DBA_CUBE_DIM_VIEWS describes the relational views of all OLAP dimensions in the database.
- USER_CUBE_DIM_VIEWS describes the relational views of the OLAP dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension, such as PRODUCT
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of a view of the cube dimension, such as PRODUCT_VIEW

See Also:

- "DBA_CUBE_DIM_VIEWS" on page 4-61
- "USER_CUBE_DIM_VIEWS" on page 5-106

ALL_CUBE_DIMENSIONALITY

ALL_CUBE_DIMENSIONALITY describes the dimension order for the OLAP cubes accessible to the current user.

Related Views

- DBA_CUBE_DIMENSIONALITY describes the dimension order for all OLAP cubes in the database.
- USER_CUBE_DIMENSIONALITY describes the dimension order for the OLAP cubes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube
CUBE_NAME	VARCHAR2(30)	NOT NULL	Name of a cube, such as UNITS_CUBE
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a dimension of the cube, such as PRODUCT
ORDER_NUM	NUMBER	NOT NULL	Order number of the dimension in the cube
IS_SPARSE	VARCHAR2(3)		Indicates whether the dimension is sparse in the cube (YES) or not (NO)

See Also:

- "DBA_CUBE_DIMENSIONALITY" on page 4-61
- "USER_CUBE_DIMENSIONALITY" on page 5-106

ALL_CUBE_DIMENSIONS

ALL_CUBE_DIMENSIONS describes the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_DIMENSIONS describes all OLAP cube dimensions in the database.
- USER_CUBE_DIMENSIONS describes the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension, such as TIME
DIMENSION_TYPE	VARCHAR2(17)		Type of the OLAP cube dimension: ■ STANDARD ■ TIME
AW_NAME	VARCHAR2(30)		Name of the analytic workspace that contains the cube dimension, such as GLOBAL
DEFAULT_HIERARCHY_NAME	VARCHAR2(30)		Name of the default hierarchy for the cube dimension, such as FISCAL
DESCRIPTION	NVARCHAR2(300)		Description of the cube dimension in the session language

See Also:

- "DBA_CUBE_DIMENSIONS" on page 4-61
- "USER_CUBE_DIMENSIONS" on page 5-106

ALL_CUBE_HIER_LEVELS

ALL_CUBE_HIER_LEVELS describes the hierarchy levels for the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_HIER_LEVELS describes the hierarchy levels for all OLAP cube dimensions in the database.

- USER_CUBE_HIER_LEVELS describes the hierarchy levels for the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension, such as TIME
HIERARCHY_NAME	VARCHAR2(30)	NOT NULL	Name of a hierarchy for the dimension, such as CALENDAR
LEVEL_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension level, such as MONTH
ORDER_NUM	NUMBER	NOT NULL	Order number of the level within the hierarchy; 0 is the top level
DESCRIPTION	NVARCHAR2(300)		Description of the level in the session language

See Also:

- "DBA_CUBE_HIER_LEVELS" on page 4-61
- "USER_CUBE_HIER_LEVELS" on page 5-106

ALL_CUBE_HIER_VIEW_COLUMNS

ALL_CUBE_HIER_VIEW_COLUMNS describes the columns of the relational hierarchy views of the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_HIER_VIEW_COLUMNS describes the columns of the relational hierarchy views of all OLAP cube dimensions in the database.
- USER_CUBE_HIER_VIEW_COLUMNS describes the columns of the relational hierarchy views of the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)		Name of a cube dimension, such as TIME
HIERARCHY_NAME	VARCHAR2(30)		Name of a hierarchy for the cube dimension, such as CALENDAR
VIEW_NAME	VARCHAR2(30)		Name of a view of the hierarchy, such as TIME_CALENDAR_VIEW
COLUMN_NAME	VARCHAR2(30)		Name of a column in the view, such as CALENDAR_QUARTER or PARENT

Column	Datatype	NULL	Description
COLUMN_TYPE	VARCHAR2(10)		Type of the column: ■ KEY - A key of the hierarchy view (that is, the hierarchy value itself) ■ PARENT - Dimension value of the parent of the current row in the view (or NULL if no parent) ■ LEVEL_NAME - Name of the level (if any) corresponding to a row in the view ■ DEPTH - Depth in the hierarchy tree of the current row in the view ■ HIER_ORDER - A column by which the results may be ordered (if present) ■ ATTRIBUTE - An attribute owned by the hierarchy ■ LEVEL - One of the level columns comprising the hierarchy
OBJECT_NAME	VARCHAR2(30)		Name of a level or attribute for the dimension

See Also:

- "DBA_CUBE_HIER_VIEW_COLUMNS" on page 4-61
- "USER_CUBE_HIER_VIEW_COLUMNS" on page 5-106

ALL_CUBE_HIER_VIEWS

ALL_CUBE_HIER_VIEWS describes the hierarchies for the OLAP cube dimensions accessible to the current user.

Related Views

- DBA_CUBE_HIER_VIEWS describes the hierarchies for all OLAP cube dimensions in the database.
- USER_CUBE_HIER_VIEWS describes the hierarchies for the OLAP cube dimensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a cube dimension, such as TIME
HIERARCHY_NAME	VARCHAR2(30)	NOT NULL	Name of a hierarchy for the cube dimension, such as CALENDAR
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of a view of the hierarchy, such as TIME_CALENDAR_VIEW

See Also:

- "DBA_CUBE_HIER_VIEWS" on page 4-61
- "USER_CUBE_HIER_VIEWS" on page 5-106

ALL_CUBE_HIERARCHIES

ALL_CUBE_HIERARCHIES describes the OLAP dimension hierarchies accessible to the current user.

Related Views

- DBA_CUBE_HIERARCHIES describes all OLAP dimension hierarchies in the database.
- USER_CUBE_HIERARCHIES describes the OLAP dimension hierarchies owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of a dimension, such as TIME
HIERARCHY_NAME	VARCHAR2(30)	NOT NULL	Name of a hierarchy for the dimension, such as CALENDAR
HIERARCHY_TYPE	VARCHAR2(5)		Type of the hierarchy: ■ LEVEL ■ VALUE
DESCRIPTION	NVARCHAR2(300)		Description of the hierarchy in the session language
IS_RAGGED	NUMBER		Indication of whether the hierarchy is ragged
IS_SKIP_LEVEL	NUMBER		Indication of whether the hierarchy is skip level

See Also:

- "DBA_CUBE_HIERARCHIES" on page 4-61
- "USER_CUBE_HIERARCHIES" on page 5-106

ALL_CUBE_MEASURES

ALL_CUBE_MEASURES describes the measures for the OLAP cubes accessible to the current user.

Related Views

- DBA_CUBE_MEASURES describes the measures for all OLAP cubes in the database.
- USER_CUBE_MEASURES describes the measures for the OLAP cubes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube
CUBE_NAME	VARCHAR2(30)	NOT NULL	Name of a cube, such as UNITS_CUBE
MEASURE_NAME	VARCHAR2(30)	NOT NULL	Name of a measure in the cube, such as SALES
OVERRIDE_SOLVE_SPEC	CLOB		Syntax text for the measure's consistent solve specification that overrides that of its cube
MEASURE_TYPE	VARCHAR2(7)		Type of the OLAP measure: ■ BASE - Base measures store the data ■ DERIVED - Derived measures calculate the data from base measures; also called calculated measures
EXPRESSION	CLOB		Expression that provides the values of the measure
DESCRIPTION	NVARCHAR2(300)		Description of the measure in the session language
DATA_TYPE	VARCHAR2(106)		Data type of the measure, such as NUMBER
DATA_LENGTH	NUMBER	NOT NULL	Length of a character data type
DATA_PRECISION	NUMBER		Precision of a numeric data type

Column	Datatype	NULL	Description
DATA_SCALE	NUMBER		Scale of a numeric data type

See Also:

- "DBA_CUBE_MEASURES" on page 4-61
- "USER_CUBE_MEASURES" on page 5-106

ALL_CUBE_VIEW_COLUMNS

ALL_CUBE_VIEW_COLUMNS describes the columns of the relational views of the OLAP cubes accessible to the current user.

Related Views

- DBA_CUBE_VIEW_COLUMNS describes the columns of relational views of all OLAP cubes in the database.
- USER_CUBE_VIEW_COLUMNS describes the columns of relational views of OLAP cubes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the cube
CUBE_NAME	VARCHAR2(30)		Name of a cube, such as UNITS_CUBE
VIEW_NAME	VARCHAR2(30)		Name of a view of the cube, such as PRODUCT_VIEW
COLUMN_NAME	VARCHAR2(30)		Name of a column in the view, such as DIM_KEY or LEVEL_NAME
COLUMN_TYPE	VARCHAR2(7)		Type of the column: ■ MEASURE ■ KEY
OBJECT_NAME	VARCHAR2(30)		Name of the measure or dimension represented in the column

See Also:

- "DBA_CUBE_VIEW_COLUMNS" on page 4-62
- "USER_CUBE_VIEW_COLUMNS" on page 5-107

ALL_CUBE_VIEWS

ALL_CUBE_VIEWS describes the relational views of the OLAP cubes accessible to the current user.

Related Views

- DBA_CUBE_VIEWS describes the relational views of all OLAP cubes in the database.
- USER_CUBE_VIEWS describes the relational views of the OLAP cubes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube

Column	Datatype	NULL	Description
CUBE_NAME	VARCHAR2(30)	NOT NULL	Name of a cube, such as UNITS_CUBE
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of a view of the cube, such as UNITS_CUBE_VIEW

See Also:

- "DBA_CUBE_VIEWS" on page 4-62
- "USER_CUBE_VIEWS" on page 5-107

ALL_CUBES

ALL_CUBES describes the OLAP cubes accessible to the current user.

Related Views

- DBA_CUBES describes all OLAP cubes in the database.
- USER_CUBES describes the OLAP cubes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the cube
CUBE_NAME	VARCHAR2(30)	NOT NULL	Name of a cube, such as UNITS_CUBE
AW_NAME	VARCHAR2(30)		Name of the analytic workspace that contains the cube, such as GLOBAL
CONSISTENT_SOLVE_SPEC	CLOB		Default aggregation rules for the cube
DESCRIPTION	NVARCHAR2(300)		Description of the cube in the session language
SPARSE_TYPE	VARCHAR2(80)		Text value indicating the type of sparsity for the OLAP cube
PRECOMPUTE_CONDITION	CLOB		Condition syntax representing the precompute condition of the OLAP cube
PRECOMPUTE_PERCENT	NUMBER		Percentage of aggregate data values that are calculated and stored during data maintenance. If the cube is partitioned, then this percentage is for the bottom partitions.
PRECOMPUTE_PERCENT_TOP	NUMBER		Percentage of aggregate data values in the top partition that are calculated and stored during data maintenance
PARTITION_DIMENSION_NAME	VARCHAR2(30)		Name of the dimension used to partition the cube, such as TIME
PARTITION_HIERARCHY_NAME	VARCHAR2(30)		Name of the dimension hierarchy used to partition the cube, such as CALENDAR
PARTITION_LEVEL_NAME	VARCHAR2(30)		Name of the level used to partition the cube, such as QUARTER

See Also:

- "DBA_CUBES" on page 4-62
- "USER_CUBES" on page 5-107

ALL_DB_LINKS

ALL_DB_LINKS describes the database links accessible to the current user.

Related Views

- DBA_DB_LINKS describes all database links in the database.
- USER_DB_LINKS describes the database links owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the database link
DB_LINK	VARCHAR2(128)	NOT NULL	Name of the database link
USERNAME	VARCHAR2(30)		Name of the user when logging in
HOST	VARCHAR2(2000)		Oracle Net connect string
CREATED	DATE	NOT NULL	Creation time of the database link

See Also:

- "DBA_DB_LINKS" on page 4-64
- "USER_DB_LINKS" on page 5-107

ALL_DEF_AUDIT_OPTS

ALL_DEF_AUDIT_OPTS contains default object-auditing options that will be applied when objects are created. The output for each column takes one of the following forms:

- -/-: no default auditing
- S/-: auditing whenever successful
- -/S: auditing whenever not successful

Column	Datatype	NULL	Description
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL / UNSUCCESSFUL
AUD	VARCHAR2(3)		Auditing AUDIT WHENEVER SUCCESSFUL / UNSUCCESSFUL
COM	VARCHAR2(3)		Auditing COMMENT WHENEVER SUCCESSFUL / UNSUCCESSFUL
DEL	VARCHAR2(3)		Auditing DELETE WHENEVER SUCCESSFUL / UNSUCCESSFUL
GRA	VARCHAR2(3)		Auditing GRANT WHENEVER SUCCESSFUL / UNSUCCESSFUL
IND	VARCHAR2(3)		Auditing INDEX WHENEVER SUCCESSFUL / UNSUCCESSFUL
INS	VARCHAR2(3)		Auditing INSERT WHENEVER SUCCESSFUL / UNSUCCESSFUL
LOC	VARCHAR2(3)		Auditing LOCK WHENEVER SUCCESSFUL / UNSUCCESSFUL
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL / UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL / UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REF	CHAR(3)		Auditing REFERENCES WHENEVER SUCCESSFUL / UNSUCCESSFUL (not used)

Column	Datatype	NULL	Description
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL / UNSUCCESSFUL
FBK	VARCHAR2(3)		Auditing FLASHBACK WHENEVER SUCCESSFUL / UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing READ WHENEVER SUCCESSFUL / UNSUCCESSFUL

ALL_DEPENDENCIES

ALL_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers accessible to the current user, including dependencies on views created without any database links. This view does not display the SCHEMAID column.

Related Views

- DBA_DEPENDENCIES describes all dependencies between objects in the database. This view does not display the SCHEMAID column.
- USER_DEPENDENCIES describes dependencies between objects in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(17)		Type of the object
REFERENCED_OWNER	VARCHAR2(30)		Owner of the referenced object (remote owner if remote object)
REFERENCED_NAME	VARCHAR2(64)		Name of the referenced object
REFERENCED_TYPE	VARCHAR2(17)		Type of the referenced object
REFERENCED_LINK_NAME	VARCHAR2(128)		Name of the link to the parent object (if remote)
SCHEMAID	NUMBER		ID of the current schema
DEPENDENCY_TYPE	VARCHAR2(4)		Indicates whether the dependency is a REF dependency (REF) or not (HARD)

ALL_DEQUEUE_QUEUES

ALL_DEQUEUE_QUEUES describes all queues on which the current user has dequeue privileges. If the user has any Advanced Queuing system privileges, such as DEQUEUE ANY QUEUE or MANAGE ANY QUEUE, then this view describes all queues in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the table in which the queue data resides
QID	NUMBER	NOT NULL	Object number of the queue
QUEUE_TYPE	VARCHAR2(20)		Type of the queue: ■ EXCEPTION_QUEUE ■ NORMAL_QUEUE

Column	Datatype	NULL	Description
MAX_RETRIES	NUMBER		Maximum number of retries allowed when dequeuing from the queue
RETRY_DELAY	NUMBER		Time interval between retries
ENQUEUE_ENABLED	VARCHAR2(7)		Indicates whether the queue is enabled for enqueue (YES) or not (NO)
DEQUEUE_ENABLED	VARCHAR2(7)		Indicates whether the queue is enabled for dequeue (YES) or not (NO)
RETENTION	VARCHAR2(40)		Time interval that processed messages are retained in the queue, or FOREVER
USER_COMMENT	VARCHAR2(50)		User-specified comment
NETWORK_NAME	VARCHAR2(512)		Network name of the queue service

ALL_DIM_ATTRIBUTES

ALL_DIM_ATTRIBUTES describes the relationship between a dimension level and a functionally dependent column. The level columns and the dependent column must be in the same table.

Related Views

- DBA_DIM_ATTRIBUTES describes all such dimension relationships in the database.
- USER_DIM_ATTRIBUTES describes all such dimension attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
ATTRIBUTE_NAME	VARCHAR2(30)		Name of the attribute
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Dependent column name
INFERRED	CHAR(1)		Indicates whether the attribute is inferred from a JOIN KEY specification (Y) or not (N)

ALL_DIM_CHILD_OF

ALL_DIM_CHILD_OF describes hierarchical relationships of 1 to *n* between the pairs of levels in the dimensions accessible to the current user.

Related Views

- DBA_DIM_CHILD_OF describes all such hierarchical relationships in the database.
- USER_DIM_CHILD_OF describes all such hierarchical attributes in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name

Column	Datatype	NULL	Description
POSITION	NUMBER	NOT NULL	Hierarchical position within this hierarchy, position 1 being the most detailed
CHILD_LEVEL_NAME	VARCHAR2(30)		Child side of 1:n relationship
JOIN_KEY_ID	VARCHAR2(40)		If non-null, then the child joins to the parent
PARENT_LEVEL_NAME	VARCHAR2(30)		Parent side of 1:n relationship in relation to the CHILD_LEVEL_NAME

ALL_DIM_HIERARCHIES

ALL_DIM_HIERARCHIES describes all dimension hierarchies accessible to the current user.

Related Views

- DBA_DIM_HIERARCHIES describes all such hierarchies in the database.
- USER_DIM_HIERARCHIES describes all such hierarchies owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
HIERARCHY_NAME	VARCHAR2(30)		Hierarchy name

ALL_DIM_JOIN_KEY

ALL_DIM_JOIN_KEY describes the joins between two dimension tables that are accessible to the current user. The join is always specified between a parent dimension level column and a child column.

Related Views

- DBA_DIM_JOIN_KEY describes all such joins in the database.
- USER_DIM_JOIN_KEY describes all such joins owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
DIM_KEY_ID	NUMBER	NOT NULL	Join key ID (unique within a dimension)
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
HIERARCHY_NAME	VARCHAR2(30)		Name of the hierarchy
CHILD_JOIN_OWNER	VARCHAR2(30)	NOT NULL	Owner of the join column table
CHILD_JOIN_TABLE	VARCHAR2(30)	NOT NULL	Name of the join column table
CHILD_JOIN_COLUMN	VARCHAR2(30)	NOT NULL	Name of the join column
CHILD_LEVEL_NAME	VARCHAR2(30)		Name of the child hierarchy level of the join key

ALL_DIM_LEVEL_KEY

ALL_DIM_LEVEL_KEY describes a column of a dimension level accessible to the current user. The position of a column within a level is specified by KEY_POSITION.

Related Views

- DBA_DIM_LEVEL_KEY describes all columns of dimension levels in the database.
- USER_DIM_LEVEL_KEY describes all columns of dimension levels owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Name of the hierarchy level
KEY_POSITION	NUMBER	NOT NULL	Ordinal position of the key column within the level
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the key column

ALL_DIM_LEVELS

ALL_DIM_LEVELS describes the dimension levels accessible to the current user. All columns of a dimension level must come from the same relation.

Related Views

- DBA_DIM_LEVELS describes all dimension levels in the database.
- USER_DIM_LEVELS describes the levels of all dimensions owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
LEVEL_NAME	VARCHAR2(30)		Unique within a dimension
NUM_COLUMNS	NUMBER		Number of columns in the level definition
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Owner of the detail object that the keys of this level come from
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Name of the table that the keys of this level come from
SKIP_WHEN_NULL	VARCHAR2(1)		Indicates whether the level is declared with the SKIP WHEN NULL clause (Y) or not (N)

ALL_DIMENSIONS

ALL_DIMENSIONS describes dimension objects accessible to the current user.

Related Views

- DBA_DIMENSIONS describes all dimensions in the database.
- USER_DIMENSIONS describes dimensions in the current user's schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the dimension

Column	Datatype	NULL	Description
DIMENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the dimension
INVALID	VARCHAR2(1)		Indicates whether the dimension is invalid (Y) or valid (N)
COMPILE_STATE	VARCHAR2(13)		Compile status of the dimension: ■ INVALID ■ NEEDS_COMPILE ■ ERROR
REVISION	NUMBER		Dimension revision level

ALL_DIRECTORIES

ALL_DIRECTORIES describes all directories accessible to the current user.

Related View

DBA_DIRECTORIES describes all directories in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the directory (always SYS)
DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the directory
DIRECTORY_PATH	VARCHAR2(4000)		Operating system pathname for the directory

ALL_ENCRYPTED_COLUMNS

ALL_ENCRYPTED_COLUMNS displays encryption algorithm information for the encrypted columns in the tables accessible to the current user.

Related Views

- DBA_ENCRYPTED_COLUMNS displays encryption algorithm information for all encrypted columns in the database.
- USER_ENCRYPTED_COLUMNS displays encryption algorithm information for the encrypted columns in the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
ENCRYPTION_ALG	VARCHAR2(29)		Encryption algorithm used to protect secrecy of data in this column: ■ 3 Key Triple DES 168 bits key ■ AES 128 bits key ■ AES 192 bits key ■ AES 256 bits key
SALT	VARCHAR2(3)		Indicates whether the column is encrypted with SALT (YES) or not (NO)

ALL_ERRORS

ALL_ERRORS describes the current errors on the stored objects accessible to the current user.

Related Views

- DBA_ERRORS describes the current errors on all stored objects in the database.
- USER_ERRORS describes the current errors on the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: ▪ VIEW ▪ PROCEDURE ▪ FUNCTION ▪ PACKAGE ▪ PACKAGE BODY ▪ TRIGGER ▪ TYPE ▪ TYPE BODY ▪ LIBRARY ▪ JAVA SOURCE ▪ JAVA CLASS ▪ DIMENSION
SEQUENCE	NUMBER	NOT NULL	Sequence number (for ordering purposes)
LINE	NUMBER	NOT NULL	Line number at which the error occurred
POSITION	NUMBER	NOT NULL	Position in the line at which the error occurred
TEXT	VARCHAR2(4000)	NOT NULL	Text of the error
ATTRIBUTE	VARCHAR2(9)		Indicates whether the error is an error (ERROR) or a warning (WARNING)
MESSAGE_NUMBER	NUMBER		Numeric error number (without any prefix)

See Also:

- "DBA_ERRORS" on page 4-68
- "USER_ERRORS" on page 5-108

ALL_EVALUATION_CONTEXT_TABLES

ALL_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_TABLES describes the tables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
TABLE_ALIAS	VARCHAR2(30)		Alias for a table in the evaluation context
TABLE_NAME	VARCHAR2(4000)		Name of the table referred to by the table alias

See Also:

- "DBA_EVALUATION_CONTEXT_TABLES" on page 4-68
- "USER_EVALUATION_CONTEXT_TABLES" on page 5-108

ALL_EVALUATION_CONTEXT_VARS

ALL_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXT_VARS describes the variables in all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
VARIABLE_NAME	VARCHAR2(30)		Name of a variable in the evaluation context
VARIABLE_TYPE	VARCHAR2(4000)		Datatype of the variable
VARIABLE_VALUE_FUNCTION	VARCHAR2(4000)		Function used to retrieve the value of the variable; NULL for variables that are not implicit
VARIABLE_METHOD_FUNCTION	VARCHAR2(228)		Function used to retrieve the result of method invocation on the variable. Such a function can speed up evaluation, if there are many simple rules that invoke the method on the variable.

See Also:

- "DBA_EVALUATION_CONTEXT_VARS" on page 4-68
- "USER_EVALUATION_CONTEXT_VARS" on page 5-109

ALL_EVALUATION_CONTEXTS

ALL_EVALUATION_CONTEXTS describes the rule evaluation contexts accessible to the current user.

Related Views

- DBA_EVALUATION_CONTEXTS describes all rule evaluation contexts in the database.
- USER_EVALUATION_CONTEXTS describes the rule evaluation contexts owned by the current user. This view does not display the EVALUATION_CONTEXT_OWNER column.

Column	Datatype	NULL	Description
EVALUATION_CONTEXT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the evaluation context
EVALUATION_CONTEXT_NAME	VARCHAR2(30)	NOT NULL	Name of the evaluation context
EVALUATION_FUNCTION	VARCHAR2(4000)		Evaluation function associated with the evaluation context, if any
EVALUATION_CONTEXT_COMMENT	VARCHAR2(4000)		Comment specified with the evaluation context, if any

See Also:

- "DBA_EVALUATION_CONTEXTS" on page 4-68
- "USER_EVALUATION_CONTEXTS" on page 5-109

ALL_EXTERNAL_LOCATIONS

ALL_EXTERNAL_LOCATIONS describes the locations (data sources) of the external tables accessible to the current user.

Related Views

- DBA_EXTERNAL_LOCATIONS describes the locations (data sources) of all external tables in the database.
- USER_EXTERNAL_LOCATIONS describes the locations (data sources) of the external tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the external table location
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the corresponding external table
LOCATION	VARCHAR2(4000)		External table location clause
DIRECTORY_OWNER	CHAR(3)		Owner of the directory containing the external table location
DIRECTORY_NAME	VARCHAR2(30)		Name of the directory containing the external table location

See Also:

- "DBA_EXTERNAL_LOCATIONS" on page 4-70
- "USER_EXTERNAL_LOCATIONS" on page 5-109

ALL_EXTERNAL_TABLES

ALL_EXTERNAL_TABLES describes the external tables accessible to the current user.

Related Views

- DBA_EXTERNAL_TABLES describes all external tables in the database.
- USER_EXTERNAL_TABLES describes the external tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the external table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the external table
TYPE_OWNER	CHAR(3)		Owner of the implementation type for the external table access driver
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the implementation type for the external table access driver
DEFAULT_DIRECTORY_OWNER	CHAR(3)		Owner of the default directory for the external table
DEFAULT_DIRECTORY_NAME	VARCHAR2(30)	NOT NULL	Name of the default directory for the external table
REJECT_LIMIT	VARCHAR2(40)		Reject limit for the external table, or UNLIMITED
ACCESS_TYPE	VARCHAR2(7)		Type of access parameters for the external table: ■ BLOB ■ CLOB
ACCESS_PARAMETERS	CLOB		Access parameters for the external table
PROPERTY	VARCHAR2(10)		Property of the external table: ■ REFERENCED - Referenced columns ■ ALL - All columns

See Also:

- "DBA_EXTERNAL_TABLES" on page 4-70
- "USER_EXTERNAL_TABLES" on page 5-109

ALL_FILE_GROUP_EXPORT_INFO

ALL_FILE_GROUP_EXPORT_INFO shows export-related information for each version accessible to the current user. There will only be information in this view for versions that have a valid Data Pump dump file.

Related Views

- DBA_FILE_GROUP_EXPORT_INFO shows export-related information for each version in the database that has a valid Data Pump dump file.
- USER_FILE_GROUP_EXPORT_INFO shows export-related information for all file groups owned by the current user. This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
VERSION_NAME	VARCHAR2(30)	NOT NULL	User-specified name for the version
VERSION	NUMBER	NOT NULL	Internal version number
EXPORT_VERSION	VARCHAR2(30)	NOT NULL	Version of exported objects

Column	Datatype	NULL	Description
PLATFORM_NAME	VARCHAR2(101)	NOT NULL	Platform on which the export was performed
EXPORT_TIME	DATE	NOT NULL	Time at which the export job was performed
EXPORT_SCN	NUMBER		SCN of the export job
SOURCE_GLOBAL_NAME	VARCHAR2(128)		Global name of the exporting database

ALL_FILE_GROUP_FILES

ALL_FILE_GROUP_FILES shows the file set for each versioned file group accessible to the current user.

Related Views

- DBA_FILE_GROUP_FILES shows the file set for each versioned group in the database.
- USER_FILE_GROUP_FILES shows the file set for each versioned group owned by the current user. This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
VERSION_NAME	VARCHAR2(30)	NOT NULL	Name of the version to which the file belongs
VERSION	NUMBER	NOT NULL	Internal version number of the file group version to which the file belongs
FILE_NAME	VARCHAR2(512)	NOT NULL	Name of the file
FILE_DIRECTORY	VARCHAR2(30)	NOT NULL	Directory object for the directory where the file is stored
FILE_TYPE	VARCHAR2(32)		User-specified file type
FILE_SIZE	NUMBER		Size of the file
FILE_BLOCK_SIZE	NUMBER		Block size for the file
COMMENTS	VARCHAR2(4000)		Comments about the file group

ALL_FILE_GROUP_TABLES

ALL_FILE_GROUP_TABLES shows information about the tables accessible to the current user that can be imported using the file set.

Related Views

- DBA_FILE_GROUP_TABLES shows information about all the tables in the database that can be imported using the file set.
- USER_FILE_GROUP_TABLES shows information about tables owned by the current user that can be imported using the file set. This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
VERSION_NAME	VARCHAR2(30)	NOT NULL	Version of the file group that contains the table
VERSION	NUMBER	NOT NULL	Internal version number

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Schema to which the table belongs
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace to which the table belongs
SCN	NUMBER		SCN at which the table was exported (available only for Streams-prepared tables)

ALL_FILE_GROUP_TABLESPACES

ALL_FILE_GROUP_TABLESPACES shows information about the transportable tablespaces present (partially or completely) in the file groups accessible to the current user (when the file groups contain dump files).

Related Views

- DBA_FILE_GROUP_TABLESPACES shows information about the transportable tablespaces present (partially or completely) in all file groups in the database (when the file groups contain dump files).
- USER_FILE_GROUP_TABLESPACES shows information about the transportable tablespaces present (partially or completely) in the file groups owned by the current user (when the file groups contain dump files). This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
VERSION_NAME	VARCHAR2(30)	NOT NULL	Version of the file group that contains the tablespace
VERSION	NUMBER	NOT NULL	Internal version number
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace

ALL_FILE_GROUP_VERSIONS

ALL_FILE_GROUP_VERSIONS shows top-level version information for the file groups accessible to the current user.

Related Views

- DBA_FILE_GROUP_VERSIONS shows top-level version information for all file groups in the database.
- USER_FILE_GROUP_VERSIONS shows top-level version information for all file groups owned by the current user. This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
VERSION_NAME	VARCHAR2(30)	NOT NULL	User-specified name for the version
VERSION	NUMBER	NOT NULL	Internal version number
CREATOR	VARCHAR2(30)	NOT NULL	User who created the version
CREATED	TIMESTAMP(6) WITH TIME ZONE	NOT NULL	Time at which the version was created

Column	Datatype	NULL	Description
COMMENTS	VARCHAR2(4000)		Comments about the file group
DEFAULT_DIRECTORY	VARCHAR2(30)		Default directory object for this version, if specified

ALL_FILE_GROUPS

ALL_FILE_GROUPS shows top-level metadata about the file groups accessible to the current user.

Related Views

- DBA_FILE_GROUPS shows top-level metadata about all file groups in the database.
- USER_FILE_GROUPS shows top-level metadata about file groups owned by the current user. This view does not display the FILE_GROUP_OWNER column.

Column	Datatype	NULL	Description
FILE_GROUP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the file group
FILE_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the file group
KEEP_FILES	VARCHAR2(1)	NOT NULL	A value of Y or N to indicate whether or not files should be deleted when a version is purged
MIN_VERSIONS	NUMBER	NOT NULL	Autopurge should not drop a version if this condition will become violated
MAX_VERSIONS	NUMBER	NOT NULL	Autopurge will drop the oldest version when this condition is violated
RETENTION_DAYS	NUMBER	NOT NULL	Autopurge will drop versions older than this if doing so does not violate MIN_VERSIONS
CREATED	TIMESTAMP(6) WITH TIME ZONE	NOT NULL	Time at which the file group was created
COMMENTS	VARCHAR2(4000)		Comments about the file group
DEFAULT_DIRECTORY	VARCHAR2(30)		Name of the default directory object

ALL_IDENTIFIERS

ALL_IDENTIFIERS displays information about the identifiers in the stored objects accessible to the current user.

Related Views

- DBA_IDENTIFIERS displays information about the identifiers in all stored objects in the database.
- USER_IDENTIFIERS displays information about the identifiers in the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the identifier
NAME	VARCHAR2(30)		Name of the identifier
SIGNATURE	VARCHAR2(32)		Signature of the identifier
TYPE	VARCHAR2(18)		Type of the identifier
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object where the identifier action occurred
OBJECT_TYPE	VARCHAR2(13)		Type of the object where the identifier action occurred

Column	Datatype	NULL	Description
USAGE	VARCHAR2(11)		Type of the identifier usage: ■ DECLARATION ■ DEFINITION ■ CALL ■ REFERENCE ■ ASSIGNMENT
USAGE_ID	NUMBER		Unique key for the identifier usage within the object
LINE	NUMBER		Line number of the identifier action
COL	NUMBER		Column number of the identifier action
USAGE_CONTEXT_ID	NUMBER		Context USAGE_ID of the identifier usage

See Also:

- "DBA_IDENTIFIERS" on page 4-135
- "USER_IDENTIFIERS" on page 5-111

ALL_IND_COLUMNS

ALL_IND_COLUMNS describes the columns of indexes on all tables accessible to the current user.

Note: For join indexes, the TABLE_NAME and TABLE_OWNER columns in this view may not match the TABLE_NAME and TABLE_OWNER columns you find in the *_INDEXES (and other similar) data dictionary views.

Related Views

- DBA_IND_COLUMNS describes the columns of indexes on all tables in the database.
- USER_IND_COLUMNS describes the columns of indexes owned by the current user and columns of indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER columns.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table or cluster
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of object type column **Note:** If you create an index on a user-defined REF column, the system creates the index on the attributes that make up the REF column. Therefore, the column names displayed in this view are the attribute names, with the REF column name as a prefix, in the following form: "REF_name"."attribute"
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within the index
COLUMN_LENGTH	NUMBER	NOT NULL	Indexed length of the column
CHAR_LENGTH	NUMBER		Maximum codepoint length of the column

Column	Datatype	NULL	Description
DESCEND	VARCHAR2(4)		Whether the column is sorted in descending order (Y/N)

ALL_IND_EXPRESSIONS

ALL_IND_EXPRESSIONS describes the expressions of function-based indexes on tables accessible to the current user.

Related Views

- DBA_IND_EXPRESSIONS describes the expressions of all function-based indexes in the database.
- USER_IND_EXPRESSIONS describes the expressions of function-based indexes on tables owned by the current user. This view does not display the INDEX_OWNER or TABLE_OWNER columns.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Index name
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table or cluster owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table or cluster name
COLUMN_EXPRESSION	LONG		Function-based index expression defining the column
COLUMN_POSITION	NUMBER	NOT NULL	Position of column or attribute within index

ALL_IND_PARTITIONS

ALL_IND_PARTITIONS describes, for each index partition accessible to the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements.

Related Views

- DBA_IND_PARTITIONS describes all index partitions in the database.
- USER_IND_PARTITIONS describes the index partitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	Description
INDEX_OWNER	VARCHAR2(30)	Owner of the index
INDEX_NAME	VARCHAR2(30)	Name of the index
COMPOSITE	VARCHAR2(3)	Indicates whether the partition belongs to a local index on a composite-partitioned table (YES) or not (NO)
PARTITION_NAME	VARCHAR2(30)	Name of the partition
SUBPARTITION_COUNT	NUMBER	If a local index on a composite-partitioned table, the number of subpartitions in the partition
HIGH_VALUE	LONG	Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	Length of partition bound value expression
PARTITION_POSITION	NUMBER	Position of the partition within the index
STATUS	VARCHAR2(8)	Whether index partition is USABLE or UNUSABLE
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace containing the partition
PCT_FREE	NUMBER	Minimum percentage of free space in a block

Column	Datatype	Description
INI_TRANS	NUMBER	Initial number of transactions
MAX_TRANS	NUMBER	Maximum number of transactions
INITIAL_EXTENT	NUMBER	Size of the initial extent in bytes
NEXT_EXTENT	NUMBER	Size of secondary extents in bytes
MIN_EXTENT	NUMBER	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	Percentage increase in extent size
FREELISTS	NUMBER	Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER	Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(7)	Logging attribute of the partition
COMPRESSION	VARCHAR2(8)	Indicates whether key compression is enabled for a partitioned index (ENABLED) or not (DISABLED); NULL for a nonpartitioned index.
BLEVEL	NUMBER	B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER	Number of leaf blocks in the index partition
DISTINCT_KEYS	NUMBER	Number of distinct keys in the index partition
AVG_LEAF_BLOCKS_PER_KEY	NUMBER	Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER	Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR	NUMBER	Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER	Number of rows returned by the ANALYZE statement
SAMPLE_SIZE	NUMBER	Sample size used in analyzing this partition
LAST_ANALYZED	DATE	Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)	Actual buffer pool for the partition
USER_STATS	VARCHAR2(3)	Indicates whether statistics were entered directly by the user (YES) or not (NO)
PCT_DIRECT_ACCESS	NUMBER	If a secondary index on index-organized table, the percentage of rows with VALID guess
GLOBAL_STATS	VARCHAR2(3)	Indicates whether statistics for the partition were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
DOMIDX_OPSTATUS	VARCHAR2(6)	???
PARAMETERS	VARCHAR2(1000)	???

ALL_IND_PENDING_STATS

ALL_IND_PENDING_STATS describes the pending statistics for tables, partitions, and subpartitions accessible to the current user.

Related Views

- DBA_IND_PENDING_STATS describes pending statistics for all tables, partitions, and subpartitions in the database.
- USER_IND_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Name of the index owner
INDEX_NAME	VARCHAR2(30)		Index name
TABLE_OWNER	VARCHAR2(30)		Table owner name
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
BLEVEL	NUMBER		Number of levels in the index
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks per key
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Clustering factor
NUM_ROWS	NUMBER		Number of rows in the index
SAMPLE_SIZE	NUMBER		Sample size
LAST_ANALYZED	DATE		Time of the last analysis

ALL_IND_STATISTICS

ALL_IND_STATISTICS displays optimizer statistics for the indexes on the tables accessible to the current user.

Related Views

- DBA_IND_STATISTICS displays optimizer statistics for all indexes in the database.
- USER_IND_STATISTICS displays optimizer statistics for the indexes on the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the index
INDEX_NAME	VARCHAR2(30)		Name of the index
TABLE_OWNER	VARCHAR2(30)		Owner of the indexed object
TABLE_NAME	VARCHAR2(30)		Name of the indexed object
PARTITION_NAME	VARCHAR2(30)		Name of the partition
PARTITION_POSITION	NUMBER		Position of the partition within the index
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition within the partition

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(12)		Type of the object: - INDEX - PARTITION - SUBPARTITION
BLEVEL	NUMBER		B-Tree level
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks per key
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks per key
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. - If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. - If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER		Number of rows in the index
AVG_CACHED_BLOCKS	NUMBER		Average number of blocks in the buffer cache
AVG_CACHE_HIT_RATIO	NUMBER		Average cache hit ratio for the object
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the index
LAST_ANALYZED	DATE		Date of the most recent time the index was analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were calculated without merging underlying partitions (YES) or not (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
STATTYPE_LOCKED	VARCHAR2(5)		Type of statistics lock
STALE_STATS	VARCHAR2(3)		Whether statistics for the object are stale or not

See Also:

- "DBA_IND_STATISTICS" on page 4-136
- "USER_IND_STATISTICS" on page 5-111

ALL_IND_SUBPARTITIONS

ALL_IND_SUBPARTITIONS describes, for each index subpartition accessible to the current user, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements.

Related Views

- DBA_IND_SUBPARTITIONS describes all index subpartitions in the database.
- USER_IND_SUBPARTITIONS describes the index subpartitions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
HIGH_VALUE	LONG		Subpartition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of subpartition bound value expression
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
STATUS	VARCHAR2(8)		Whether index partition is usable or not
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of process freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
COMPRESSION	VARCHAR2(8)		Indicates whether this subpartition is compressed (ENABLED) or not (DISABLED)
BLEVEL	NUMBER		B-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index partition
AVG_LEAF_BLOCKS_PER_KEY	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks.
NUM_ROWS	NUMBER		Number of rows in this index subpartition
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this subpartition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed

Column	Datatype	NULL	Description
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the subpartition
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition statistics were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO)

ALL_INDEXES

ALL_INDEXES describes the indexes on the tables accessible to the current user. To gather statistics for this view and the related views DBA_INDEXES and USER_INDEXES, use the SQL ANALYZE statement.

Related Views

- DBA_INDEXES describes all indexes in the database.
- USER_INDEXES describes the indexes owned by the current user. This view does not display the OWNER column.

Note: Column names followed by an asterisk are populated only if you collect statistics on the index using the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
INDEX_TYPE	VARCHAR2(27)		Type of the index: ■ NORMAL ■ BITMAP ■ FUNCTION-BASED NORMAL ■ FUNCTION-BASED BITMAP ■ DOMAIN
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the indexed object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
TABLE_TYPE	CHAR(5)		Type of the indexed object (for example, TABLE, CLUSTER)
UNIQUENESS	VARCHAR2(9)		Indicates whether the index is UNIQUE or NONUNIQUE
COMPRESSION	VARCHAR2(8)		Indicates whether index compression is enabled (ENABLED) or not (DISABLED)
PREFIX_LENGTH	NUMBER		Number of columns in the prefix of the compression key
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the index
INI_TRANS	NUMBER		Initial number of transactions
MAX_TRANS	NUMBER		Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent
NEXT_EXTENT	NUMBER		Size of secondary extents
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment

Column	Datatype	NULL	Description
PCT_INCREASE	NUMBER		Percentage increase in extent size
PCT_THRESHOLD	NUMBER		Threshold percentage of block space allowed per index entry
INCLUDE_COLUMN	NUMBER		Column ID of the last column to be included in index-organized table primary key (non-overflow) index. This column maps to the COLUMN_ID column of the *_TAB_COLUMNS data dictionary views.
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
PCT_FREE	NUMBER		Minimum percentage of free space in a block
LOGGING	VARCHAR2(3)		Logging information
BLEVEL*	NUMBER		B*-Tree level: depth of the index from its root block to its leaf blocks. A depth of 0 indicates that the root block and leaf block are the same.
LEAF_BLOCKS*	NUMBER		Number of leaf blocks in the index
DISTINCT_KEYS*	NUMBER		Number of distinct indexed values. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is the same as the number of rows in the table (USER_TABLES.NUM_ROWS)
AVG_LEAF_BLOCKS_PER_KEY*	NUMBER		Average number of leaf blocks in which each distinct value in the index appears, rounded to the nearest integer. For indexes that enforce UNIQUE and PRIMARY KEY constraints, this value is always 1.
AVG_DATA_BLOCKS_PER_KEY*	NUMBER		Average number of data blocks in the table that are pointed to by a distinct value in the index rounded to the nearest integer. This statistic is the average number of data blocks that contain rows that contain a given value for the indexed columns.
CLUSTERING_FACTOR*	NUMBER		Indicates the amount of order of the rows in the table based on the values of the index. ■ If the value is near the number of blocks, then the table is very well ordered. In this case, the index entries in a single leaf block tend to point to rows in the same data blocks. ■ If the value is near the number of rows, then the table is very randomly ordered. In this case, it is unlikely that index entries in the same leaf block point to rows in the same data blocks. For bitmap indexes, this column is not applicable and is not used.
STATUS	VARCHAR2(8)		Indicates whether a nonpartitioned index is VALID or UNUSABLE
NUM_ROWS	NUMBER		Number of rows in the index
SAMPLE_SIZE	NUMBER		Size of the sample used to analyze the index
LAST_ANALYZED	DATE		Date on which this index was most recently analyzed
DEGREE	VARCHAR2(40)		Number of threads per instance for scanning the index
INSTANCES	VARCHAR2(40)		Number of instances across which the indexes to be scanned
PARTITIONED	VARCHAR2(3)		Indicates whether the index is partitioned (YES) or not (NO)
TEMPORARY	VARCHAR2(1)		Indicates whether the index is on a temporary table
GENERATED	VARCHAR2(1)		Indicates whether the name of the index is system generated (Y) or not (N)

Column	Datatype	NULL	Description
SECONDARY	VARCHAR2(1)		Indicates whether the index is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y) or not (N)
BUFFER_POOL	VARCHAR2(7)		Name of the default buffer pool to be used for the index blocks
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION - Rows are preserved for the duration of the session ■ SYS$TRANSACTION - Rows are deleted after COMMIT Null for a permanent table
PCT_DIRECT_ACCESS	NUMBER		For a secondary index on an index-organized table, the percentage of rows with VALID guess
ITYP_OWNER	VARCHAR2(30)		For a domain index, the owner of the indextype
ITYP_NAME	VARCHAR2(30)		For a domain index, the name of the indextype
PARAMETERS	VARCHAR2(1000)		For a domain index, the parameter string
GLOBAL_STATS	VARCHAR2(3)		For partitioned indexes, indicates whether statistics were collected by analyzing the index as a whole (YES) or were estimated from statistics on underlying index partitions and subpartitions (NO)
DOMIDX_STATUS	VARCHAR2(12)		Status of the domain index: ■ NULL - Index is not a domain index ■ VALID - Index is a valid domain index ■ IDXTYP_INVLD - Indextype of the domain index is invalid
DOMIDX_OPSTATUS	VARCHAR2(6)		Status of the operation on the domain index: ■ NULL - Index is not a domain index ■ VALID - Operation performed without errors ■ FAILED - Operation failed with an error
FUNCIDX_STATUS	VARCHAR2(8)		Status of a function-based index: ■ NULL - Index is not a function-based index ■ ENABLED - Function-based index is enabled ■ DISABLED - Function-based index is disabled
JOIN_INDEX	VARCHAR2(3)		Indicates whether the index is a join index (YES) or not (NO)
IOT_REDUNDANT_PKEY_ELIM	VARCHAR2(3)		Indicates whether redundant primary key columns are eliminated from secondary indexes on index-organized tables (YES) or not (NO)
DROPPED	VARCHAR2(3)		Indicates whether the index has been dropped and is in the recycle bin (YES) or not (NO); NULL for partitioned tables
VISIBILITY	VARCHAR2(10)		Whether the index is VISIBLE or INVISIBLE to the Optimizer
DOMIDX_MANAGEMENT	VARCHAR2(14)		If this is a domain index, then whether it is system-managed or user-managed

ALL_INDEXTYPE_ARRAYTYPES

ALL_INDEXTYPE_ARRAYTYPES displays information about the array types specified by the indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_ARRAYTYPES displays information about the array types specified by all indextypes in the database.
- USER_INDEXTYPE_ARRAYTYPES displays information about the array types specified by the indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
BASE_TYPE_SCHEMA	VARCHAR2(30)		Name of the base type schema
BASE_TYPE_NAME	VARCHAR2(30)		Name of the base type name
BASE_TYPE	VARCHAR2(30)		Datatype of the base type
ARRAY_TYPE_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the array type schema
ARRAY_TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the array type name

See Also:

- "DBA_INDEXTYPE_ARRAYTYPES" on page 4-136
- "USER_INDEXTYPE_ARRAYTYPES" on page 5-112

ALL_INDEXTYPE_COMMENTS

ALL_INDEXTYPE_COMMENTS displays comments for the user-defined indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_COMMENTS displays comments for all user-defined indextypes in the database.
- USER_INDEXTYPE_COMMENTS displays comments for the user-defined indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the user-defined indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the user-defined indextype
COMMENTS	VARCHAR2(4000)		Comment for the user-defined indextype

See Also:

- "DBA_INDEXTYPE_COMMENTS" on page 4-136
- "USER_INDEXTYPE_COMMENTS" on page 5-112

ALL_INDEXTYPE_OPERATORS

ALL_INDEXTYPE_OPERATORS lists all operators supported by indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPE_OPERATORS lists all operators supported by indextypes in the database.

- USER_INDEXTYPE_OPERATORS lists all operators supported by indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
OPERATOR_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the operator schema
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator for which the indextype is defined
BINDING#	NUMBER	NOT NULL	Binding number associated with the operator

See Also:

- "DBA_INDEXTYPE_OPERATORS" on page 4-136
- "USER_INDEXTYPE_OPERATORS" on page 5-112

ALL_INDEXTYPES

ALL_INDEXTYPES displays information about the indextypes accessible to the current user.

Related Views

- DBA_INDEXTYPES displays information about all indextypes in the database.
- USER_INDEXTYPES displays information about the indextypes owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the indextype
INDEXTYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype
IMPLEMENTATION_SCHEMA	VARCHAR2(30)	NOT NULL	Name of the schema for the indextype implementation (that is, containing the indextype operators)
IMPLEMENTATION_NAME	VARCHAR2(30)	NOT NULL	Name of the indextype implementation type
INTERFACE_VERSION	NUMBER		Version of the indextype interface
IMPLEMENTATION_VERSION	NUMBER	NOT NULL	Version of the indextype implementation
NUMBER_OF_OPERATORS	NUMBER		Number of operators associated with the indextype
PARTITIONING	VARCHAR2(10)		Kinds of local partitioning supported by the indextype: ▪ NONE ▪ RANGE ▪ HASH ▪ HASH,RANGE
ARRAY_DML	VARCHAR2(3)		Indicates whether the indextype supports array DML (YES) or not (NO)
MAINTENANCE_TYPE	VARCHAR2(14)		Whether the indextype is system-managed or user-managed

See Also:

- "DBA_INDEXTYPES" on page 4-136
- "USER_INDEXTYPES" on page 5-112

ALL_INTERNAL_TRIGGERS

ALL_INTERNAL_TRIGGERS describes internal triggers on tables accessible to the current user. Internal triggers are internal pieces of code executed when a particular flag is set for a table. This view does not display the OWNER_NAME column.

Related Views

- DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database.
- USER_INTERNAL_TRIGGERS describes all internal triggers on tables owned by the current user. This view does not display the OWNER_NAME column.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the table on which the trigger is defined
OWNER_NAME	VARCHAR2(30)		Owner of the table
INTERNAL_TRIGGER_TYPE	VARCHAR2(19)		Indicates the type of internal trigger on the table

ALL_JAVA_ARGUMENTS

ALL_JAVA_ARGUMENTS displays argument information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_ARGUMENTS displays argument information about all stored Java classes in the database.
- USER_JAVA_ARGUMENTS displays argument information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the hosting method of the argument
METHOD_NAME	VARCHAR2(4000)		Name of the hosting method of the argument
ARGUMENT_POSITION	NUMBER		Position of the argument, starting from 0
ARRAY_DEPTH	NUMBER		Array depth of the type of the argument
BASE_TYPE	VARCHAR2(7)		Base type of the type of the argument: ■ int ■ long ■ float ■ double ■ boolean ■ byte ■ char ■ short ■ class
ARGUMENT_CLASS	VARCHAR2(4000)		Actual class name of the argument if the base type is class

See Also:

- "DBA_JAVA_ARGUMENTS" on page 4-137
- "USER_JAVA_ARGUMENTS" on page 5-112

ALL_JAVA_CLASSES

ALL_JAVA_CLASSES displays class level information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_CLASSES displays class level information about all stored Java classes in the database.
- USER_JAVA_CLASSES displays class level information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
MAJOR	NUMBER		Major version number of the Java class, as defined in the JVM specification
MINOR	NUMBER		Minor version number of the Java class, as defined in the JVM specification
KIND	VARCHAR2(9)		Indicates whether the stored object is a Java class (CLASS) or a Java interface (INTERFACE)
ACCESSIBILITY	VARCHAR2(6)		Accessibility of the Java class
IS_INNER	VARCHAR2(3)		Indicates whether this Java class is an inner class (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether this Java class is an abstract class (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether this Java class is a final class (YES) or not (NO)
IS_DEBUG	VARCHAR2(3)		Indicates whether this Java class contains debug information (YES) or not (NO)
SOURCE	VARCHAR2(4000)		Source designation of the Java class
SUPER	VARCHAR2(4000)		Super class of this Java class
OUTER	VARCHAR2(4000)		Outer class of this Java class if this Java class is an inner class

See Also:

- "DBA_JAVA_CLASSES" on page 4-137
- "USER_JAVA_CLASSES" on page 5-113

ALL_JAVA_COMPILER_OPTIONS

ALL_JAVA_COMPILER_OPTIONS displays information about the native compiler options accessible to the current user.

Related Views

- DBA_JAVA_COMPILER_OPTIONS displays information about all native compiler options in the database.
- USER_JAVA_COMPILER_OPTIONS displays information about the native compiler options owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the native compiler option
OPTION_NAME	VARCHAR2(64)		Name of the native compiler option
VALUE	VARCHAR2(4000)		Value of the native compiler option

See Also:

- "DBA_JAVA_COMPILER_OPTIONS" on page 4-137
- "USER_JAVA_COMPILER_OPTIONS" on page 5-113

ALL_JAVA_DERIVATIONS

ALL_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes accessible to the current user.

Related Views

- DBA_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for all Java classes in the database.
- USER_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java source object
SOURCE_NAME	VARCHAR2(4000)		Name of the Java source object
CLASS_INDEX	NUMBER		Index of the derived Java class object
CLASS_NAME	VARCHAR2(4000)		Name of the derived Java class object

See Also:

- "DBA_JAVA_DERIVATIONS" on page 4-137
- "USER_JAVA_DERIVATIONS" on page 5-113

ALL_JAVA_FIELDS

ALL_JAVA_FIELDS displays field information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_FIELDS displays field information about all stored Java classes in the database.

- USER_JAVA_FIELDS displays field information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
FIELD_INDEX	NUMBER		Index of the field
FIELD_NAME	VARCHAR2(4000)		Name of the field identified by the FIELD_INDEX column
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the field: ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the field is a static field (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the field is a final field (YES) or not (NO)
IS_VOLATILE	VARCHAR2(3)		Indicates whether the field is volatile (YES) or not (NO)
IS_TRANSIENT	VARCHAR2(3)		Indicates whether the field is transient (YES) or not (NO)
ARRAY_DEPTH	NUMBER		Array depth of the type of the field
BASE_TYPE	VARCHAR2(7)		Base type of the type of the field: ■ int ■ long ■ float ■ double ■ boolean ■ byte ■ char ■ short ■ class
FIELD_CLASS	VARCHAR2(4000)		Actual class name of the base object if the base type is class

See Also:

- "DBA_JAVA_FIELDS" on page 4-137
- "USER_JAVA_FIELDS" on page 5-113

ALL_JAVA_IMPLEMENTS

ALL_JAVA_IMPLEMENTS describes interfaces implemented by the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_IMPLEMENTS describes interfaces implemented by all stored Java classes in the database.
- USER_JAVA_IMPLEMENTS describes interfaces implemented by the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INTERFACE_INDEX	NUMBER		Index of the interfaces implemented by the Java class
INTERFACE_NAME	VARCHAR2(4000)		Name of the interface identified by the INTERFACE_INDEX column

See Also:

- "DBA_JAVA_IMPLEMENTS" on page 4-137
- "USER_JAVA_IMPLEMENTS" on page 5-113

ALL_JAVA_INNERS

ALL_JAVA_INNERS displays information about inner classes referred to by the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_INNERS displays information about inner classes referred to by all stored Java classes in the database.
- USER_JAVA_INNERS displays information about inner classes referred to by the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INNER_INDEX	NUMBER		Index of the referred inner class
SIMPLE_NAME	VARCHAR2(4000)		Simple name of the referred inner class
FULL_NAME	VARCHAR2(4000)		Full name of the referred inner class
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the referred inner class: ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the referred inner class is declared static in the source file (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the referred inner class is declared final in the source file (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether the referred inner class is declared abstract in the source file (YES) or not (NO)
IS_INTERFACE	VARCHAR2(3)		Indicates whether the referred inner class is declared interface in the source file (YES) or not (NO)

See Also:

- "DBA_JAVA_INNERS" on page 4-137
- "USER_JAVA_INNERS" on page 5-113

ALL_JAVA_LAYOUTS

ALL_JAVA_LAYOUTS displays class layout information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_LAYOUTS displays class layout information about all stored Java classes in the database.
- USER_JAVA_LAYOUTS displays class layout information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
INTERFACES	NUMBER		Number of interfaces that this Java class implements
INNER_CLASSES	NUMBER		Number of inner classes that this Java class contains
FIELDS	NUMBER		Number of locally declared fields that this Java class contains
STATIC_FIELDS	NUMBER		Number of locally declared static fields that this Java class contains
METHODS	NUMBER		Number of locally declared methods that this Java class contains
STATIC_METHODS	NUMBER		Number of locally declared static methods that this Java class contains
NATIVE_METHODS	NUMBER		Number of locally declared native methods that this Java class contains

See Also:

- "DBA_JAVA_LAYOUTS" on page 4-137
- "USER_JAVA_LAYOUTS" on page 5-113

ALL_JAVA_METHODS

ALL_JAVA_METHODS displays method information about the stored Java classes accessible to the current user.

Related Views

- DBA_JAVA_METHODS displays method information about all stored Java classes in the database.
- USER_JAVA_METHODS displays method information about the stored Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the method
METHOD_NAME	VARCHAR2(4000)		Name of the method identified by the METHOD_INDEX column

Column	Datatype	NULL	Description
ACCESSIBILITY	VARCHAR2(9)		Accessibility of the method: ■ PUBLIC ■ PRIVATE ■ PROTECTED
IS_STATIC	VARCHAR2(3)		Indicates whether the method is a static method (YES) or not (NO)
IS_FINAL	VARCHAR2(3)		Indicates whether the method is a final method (YES) or not (NO)
IS_SYNCHRONIZED	VARCHAR2(3)		Indicates whether the method is a synchronized method (YES) or not (NO)
IS_NATIVE	VARCHAR2(3)		Indicates whether the method is a native method (YES) or not (NO)
IS_ABSTRACT	VARCHAR2(3)		Indicates whether the method is an abstract method (YES) or not (NO)
IS_STRICT	VARCHAR2(3)		Indicates whether the method is a strict method (YES) or not (NO)
ARGUMENTS	NUMBER		Number of arguments of the method
THROWS	NUMBER		Number of exceptions thrown by the method
ARRAY_DEPTH	NUMBER		Array depth of the return type of the method
BASE_TYPE	VARCHAR2(7)		Base type of the return type of the method: ■ int ■ long ■ float ■ double ■ boolean ■ byte ■ char ■ short ■ class ■ void
RETURN_CLASS	VARCHAR2(4000)		Actual class name of the return value if the base type is class
IS_COMPILED	VARCHAR2(3)		Indicates whether the Java method has been natively compiled by the JIT compiler (YES) or not (NO)

See Also:

- "DBA_JAVA_METHODS" on page 4-138
- "USER_JAVA_METHODS" on page 5-114

ALL_JAVA_NCOMPS

ALL_JAVA_NCOMPS displays ncomp-related information about the Java classes accessible to the current user.

Related Views

- DBA_JAVA_NCOMPS displays ncomp-related information about all Java classes in the database.
- USER_JAVA_NCOMPS displays ncomp-related information about the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class object
NAME	VARCHAR2(4000)		Name of the Java class object
SOURCE	VARCHAR2(4000)		ncomp source shown in this row
INITIALIZER	VARCHAR2(4000)		ncomp initializer shown in this row
LIBRARYFILE	VARCHAR2(4000)		ncomp library file shown in this row
LIBRARY	VARCHAR2(4000)		ncomp library shown in this row

See Also:

- "DBA_JAVA_NCOMPS" on page 4-138
- "USER_JAVA_NCOMPS" on page 5-114

ALL_JAVA_RESOLVERS

ALL_JAVA_RESOLVERS displays information about resolvers of the Java classes accessible to the current user.

Related Views

- DBA_JAVA_RESOLVERS displays information about resolvers of all Java classes in the database.
- USER_JAVA_RESOLVERS displays information about resolvers of the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class object
NAME	VARCHAR2(4000)		Name of the Java class object
TERM_INDEX	NUMBER		Index of the resolver term in this row
PATTERN	VARCHAR2(4000)		Resolver pattern of the resolver term identified by the TERM_INDEX column
SCHEMA	VARCHAR2(64)		Resolver schema of the resolver term identified by the TERM_INDEX column

See Also:

- "DBA_JAVA_RESOLVERS" on page 4-138
- "USER_JAVA_RESOLVERS" on page 5-114

ALL_JAVA_THROWS

ALL_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes accessible to the current user.

Related Views

- DBA_JAVA_THROWS displays information about exceptions thrown from methods of all Java classes in the database.
- USER_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Java class
NAME	VARCHAR2(4000)		Name of the Java class
METHOD_INDEX	NUMBER		Index of the throwing method of the exception
METHOD_NAME	VARCHAR2(4000)		Name of the throwing method of the exception
EXCEPTION_INDEX	NUMBER		Index of the exception
EXCEPTION_CLASS	VARCHAR2(4000)		Class of the exception

See Also:

- "DBA_JAVA_THROWS" on page 4-139
- "USER_JAVA_THROWS" on page 5-114

ALL_JOBS

ALL_JOBS is a synonym for USER_JOBS.

See Also: "USER_JOBS" on page 5-114

ALL_JOIN_IND_COLUMNS

ALL_JOIN_IND_COLUMNS describes the join conditions of bitmap join indexes to which you have access. Bitmap join indexes are indexes built on a child table with an index key containing columns from associated parent tables, where all of the tables are connected through join conditions. There is one row for each join condition.

Related Views

- DBA_JOIN_IND_COLUMNS describes all join conditions exiting in the database.
- USER_JOIN_IND_COLUMNS describes join conditions owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	Description
INDEX_OWNER	VARCHAR2(30)	Bitmap join index owner
INDEX_NAME	VARCHAR2(30)	Name of bitmap join index
INNER_TABLE_OWNER	VARCHAR2(30)	Fact table owner
INNER_TABLE_NAME	VARCHAR2(30)	Name of the fact table
OUTER_TABLE_OWNER	VARCHAR2(30)	Dimension table owner
OUTER_TABLE_NAME	VARCHAR2(30)	Name of the dimension table
COL_NAME_INNER	VARCHAR2(4000)	Name of the fact table join column
COL_NAME_OUTER	VARCHAR2(4000)	Name of the dimension table join column

ALL_LIBRARIES

ALL_LIBRARIES describes the libraries accessible to the current user.

Related Views

- DBA_LIBRARIES describes all libraries in the database.

- USER_LIBRARIES describes the libraries owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the library
LIBRARY_NAME	VARCHAR2(30)	NOT NULL	Library name
FILE_SPEC	VARCHAR2(2000)		Operating system file specification associated with the library
DYNAMIC	VARCHAR2(1)		Indicates whether the library is dynamically loadable (Y) or not (N)
STATUS	VARCHAR2(7)		Status of the library: ■ N/A ■ VALID ■ INVALID

See Also:

- "DBA_LIBRARIES" on page 4-140
- "USER_LIBRARIES" on page 5-114

ALL_LOB_PARTITIONS

ALL_LOB_PARTITIONS displays LOB partitions contained in tables accessible to the current user.

Related Views

- DBA_LOB_PARTITIONS describes all LOB partitions in the database.
- USER_LOB_PARTITIONS describes the LOB partitions owned by the current user. This view does not display the TABLE_OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column
LOB_NAME	VARCHAR2(30)		Name of the partitioned LOB item
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
LOB_PARTITION_NAME	VARCHAR2(30)		Name of the LOB data partition
LOB_INDPART_NAME	VARCHAR2(30)		Name of the corresponding LOB index partition
PARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
COMPOSITE	VARCHAR2(3)		Whether the partition is composite (YES | NO)
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER		Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(10)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)
IN_ROW	VARCHAR2(3)		Whether the STORAGE IN ROW attribute is enabled for the LOB data partition
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB data partition

Column	Datatype	NULL	Description
INITIAL_EXTENT	VARCHAR2(40)		Size in bytes of the initial extent of the LOB data partition
NEXT_EXTENT	VARCHAR2(40)		Size in bytes of secondary extents of the LOB data partition
MIN_EXTENTS	VARCHAR2(40)		Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	VARCHAR2(40)		Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	VARCHAR2(40)		Percentage increase in extent size for the LOB data partition
FREELISTS	VARCHAR2(40)		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	VARCHAR2(40)		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(7)		Logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the LOB partition blocks
ENCRYPT	VARCHAR2(3)		Whether or not the LOB is encrypted
COMPRESSION	VARCHAR2(6)		The level of compression used for this LOB
DEDUPLICATION	VARCHAR2(15)		The kind of deduplication used for this LOB
SECUREFILE	VARCHAR2(3)		Indicates whether the LOB is a SecureFile LOB (YES) or not (NO)

ALL_LOB_SUBPARTITIONS

ALL_LOB_SUBPARTITIONS displays partition-level attributes of LOB data subpartitions accessible to the current user.

Related Views

- DBA_LOB_SUBPARTITIONS describes all LOB subpartitions in the database.
- USER_LOB_SUBPARTITIONS describes the LOB subpartitions owned by the current user. This view does not display the TABLE_OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB item
LOB_PARTITION_NAME	VARCHAR2(30)		Name of LOB data partition to which this LOB data subpartition belongs
SUBPARTITION_NAME	VARCHAR2(30)		Name of the table subpartition to which this LOB subpartition corresponds
LOB_SUBPARTITION_NAME	VARCHAR2(30)		Name of the LOB subpartition
LOB_INDSUBPART_NAME	VARCHAR2(30)		Name of corresponding LOB index subpartition
SUBPARTITION_POSITION	NUMBER		Position of the LOB data partition within the LOB item
CHUNK	NUMBER		Value of the CHUNK attribute of the LOB data partition
PCTVERSION	NUMBER	NOT NULL	Value of the PCTVERSION attribute of the LOB data partition
CACHE	VARCHAR2(10)		Whether and how the cluster is to be cached in the buffer cache (CACHE, NOCACHE, CACHEREADS)

Column	Datatype	NULL	Description
IN_ROW	VARCHAR2(3)		Whether the STORAGE IN ROW attribute of the LOB data partition is enabled
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the LOB data partition
INITIAL_EXTENT	NUMBER		Size in bytes of the initial extent for the LOB data partition
NEXT_EXTENT	NUMBER		Size in bytes of secondary extents for the LOB data partition
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents allowed in the segment of the LOB data partition
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extents allowed in the segment of the LOB data partition
PCT_INCREASE	NUMBER		Percentage increase in extent size for the LOB data partition
FREELISTS	NUMBER		Number of process freelists allocated in the segment of the LOB data partition
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in the segment of the LOB data partition
LOGGING	VARCHAR2(3)		Logging attribute of the LOB data partition
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for the LOB data partition blocks
ENCRYPT	VARCHAR2(3)		Whether or not the LOB is encrypted
COMPRESSION	VARCHAR2(6)		The level of compression used for this LOB
DEDUPLICATION	VARCHAR2(15)		The kind of deduplication used for this LOB
SECUREFILE	VARCHAR2(3)		Indicates whether the LOB is a SecureFile LOB (YES) or not (NO)

ALL_LOB_TEMPLATES

ALL_LOB_TEMPLATES describes the LOB subpartition templates accessible to the current user.

Related Views

- DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.
- USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
LOB_COL_NAME	VARCHAR2(4000)		Name of the LOB column
SUBPARTITION_NAME	VARCHAR2(34)	NOT NULL	Name of the subpartition
LOB_SEGMENT_NAME	VARCHAR2(34)	NOT NULL	Name of the LOB segment
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name of the subpartition

See Also:

- "DBA_LOB_TEMPLATES" on page 4-141
- "USER_LOB_TEMPLATES" on page 5-115

ALL_LOBS

ALL_LOBS displays the large objects (LOBs) contained in tables accessible to the current user. LOBs include binary large objects (BLOBs) and character large objects (CLOBs). Binary files (BFILEs) are stored outside the database, so they are not displayed by this view or the related views.

Related Views

- DBA_LOBS describes all LOBs in the database.
- USER_LOBS describes the LOBs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object containing the LOB
TABLE_NAME	VARCHAR2(30)		Name of the object containing the LOB
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column or attribute
SEGMENT_NAME	VARCHAR2(30)		Name of the LOB segment
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the LOB segment
INDEX_NAME	VARCHAR2(30)		Name of the LOB index
CHUNK	NUMBER		Size (in bytes) of the LOB chunk as a unit of allocation or manipulation
PCTVERSION	NUMBER		Maximum percentage of the LOB space used for versioning
RETENTION	NUMBER		Maximum time duration for versioning of the LOB space
FREEPOOLS	NUMBER		Number of freepools for this LOB segment
CACHE	VARCHAR2(10)		Indicates whether and how the cluster is to be cached in the buffer cache (YES, NO, CACHEREADS)
LOGGING	VARCHAR2(7)		Indicates whether changes to the LOB are logged (YES) or not (NO)
ENCRYPT	VARCHAR2(3)		Whether or not the LOB is encrypted
COMPRESSION	VARCHAR2(6)		The level of compression used for this LOB
DEDUPLICATION	VARCHAR2(15)		The kind of deduplication used for this LOB
IN_ROW	VARCHAR2(3)		Indicates whether some of the LOBs are stored inline with the base row (YES) or not (NO). For partitioned objects, refer to the *_LOB_PARTITIONS and *_PART_LOBS views.
FORMAT	VARCHAR2(15)		Indicates whether the LOB storage format is dependent on the endianness of the platform: ■ NOT APPLICABLE ■ ENDIAN SPECIFIC ■ ENDIAN NEUTRAL
PARTITIONED	VARCHAR2(3)		Indicates whether the LOB column is in a partitioned table (YES) or not (NO)
SECUREFILE	VARCHAR2(3)		Indicates whether the LOB is a SecureFile LOB (YES) or not (NO)

ALL_LOG_GROUP_COLUMNS

ALL_LOG_GROUP_COLUMNS describes columns that are accessible to the current user and that are specified in log groups.

Related Views

- DBA_LOG_GROUP_COLUMNS describes all columns in the database that are specified in log groups.
- USER_LOG_GROUP_COLUMNS describes columns that are owned by the current user and that are specified in log groups.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the log group definition
LOG_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the log group definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table in which the log group is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column or attribute of the object type column specified in the log group definition
POSITION	NUMBER		Original position of the column or attribute in the definition of the object
LOGGING_PROPERTY	VARCHAR2(6)		Indicates whether the column or attribute would be supplementally logged (LOG) or not (NO LOG)

ALL_LOG_GROUPS

ALL_LOG_GROUPS describes the log group definitions on the tables accessible to the current user.

Related Views

- DBA_LOG_GROUPS describes the log group definitions on all tables in the database.
- USER_LOG_GROUPS describes the log group definitions on the tables owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the log group definition
LOG_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the log group definition
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table on which the log group is defined
LOG_GROUP_TYPE	VARCHAR2(19)		Type of the log group: ■ PRIMARY KEY LOGGING ■ UNIQUE KEY LOGGING ■ FOREIGN KEY LOGGING ■ ALL COLUMN LOGGING ■ USER LOG GROUP
ALWAYS	VARCHAR2(11)		Y indicates the log group is logged any time a row is updated; N indicates the log group is logged any time a member column is updated.
GENERATED	VARCHAR2(14)		Indicates whether the name of the supplemental log group was system generated (GENERATED NAME) or not (USER NAME)

ALL_MEASURE_FOLDER_CONTENTS

ALL_MEASURE_FOLDER_CONTENTS describes the contents of the OLAP measure folders accessible to the current user.

Related Views

- DBA_MEASURE_FOLDER_CONTENTS describes the contents of all OLAP measure folders in the database.
- USER_MEASURE_FOLDER_CONTENTS describes the contents of the OLAP measure folders owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the measure folder
MEASURE_FOLDER_NAME	VARCHAR2(30)	NOT NULL	Name of a measure folder
CUBE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the cube
CUBE_NAME	VARCHAR2(30)	NOT NULL	Name of a cube included in the measure folder
MEASURE_NAME	VARCHAR2(30)	NOT NULL	Name of a measure in the cube
ORDER_NUM	NUMBER	NOT NULL	Order number of the measure in the folder

See Also:

- "DBA_MEASURE_FOLDER_CONTENTS" on page 4-149
- "USER_MEASURE_FOLDER_CONTENTS" on page 5-115

ALL_MEASURE_FOLDERS

ALL_MEASURE_FOLDERS describes the OLAP measure folders accessible to the current user.

Related Views

- DBA_MEASURE_FOLDERS describes all OLAP measure folders in the database.
- USER_MEASURE_FOLDERS describes the OLAP measure folders owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the measure folder
MEASURE_FOLDER_NAME	VARCHAR2(30)	NOT NULL	Name of a measure folder
DESCRIPTION	NVARCHAR2(300)		Description of the measure folder in the session language

See Also:

- "DBA_MEASURE_FOLDERS" on page 4-149
- "USER_MEASURE_FOLDERS" on page 5-115

ALL_METHOD_PARAMS

ALL_METHOD_PARAMS describes the method parameters of the object types accessible to the current user.

Related Views

- DBA_METHOD_PARAMS describes the method parameters of all object types in the database.

- USER_METHOD_PARAMS describes the method parameters of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
PARAM_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAM_NO	NUMBER	NOT NULL	Parameter number (position)
PARAM_MODE	VARCHAR2(6)		Mode of the parameter (IN, OUT, IN OUT)
PARAM_TYPE_MOD	VARCHAR2(7)		Whether this parameter is a REF to another object
PARAM_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the parameter
PARAM_TYPE_NAME	VARCHAR2(30)		Name of the type of the parameter
CHARACTER_SET_NAME	VARCHAR2(44)		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

See Also:

- "DBA_METHOD_PARAMS" on page 4-149
- "USER_METHOD_PARAMS" on page 5-116

ALL_METHOD_RESULTS

ALL_METHOD_RESULTS describes the method results of the object types accessible to the current user.

Related Views

- DBA_METHOD_RESULTS describes the method results of all object types in the database.
- USER_METHOD_RESULTS describes the method results of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	For an overloaded method, a number distinguishing this method from others of the same. Do not confuse this number with the object ID.
RESULT_TYPE_MOD	VARCHAR2(7)	NOT NULL	Whether this result is a REF to another object
RESULT_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the result
RESULT_TYPE_NAME	VARCHAR2(30)		Name of the type of the result
CHARACTER_SET_NAME	VARCHAR2(44)		Whether the character set or the method is fixed-length character set (CHAR_CS) or fixed-length national character set (NCHAR_CS), or a particular character set specified by the user

See Also:

- "DBA_METHOD_RESULTS" on page 4-150
- "USER_METHOD_RESULTS" on page 5-116

ALL_MINING_MODEL_ATTRIBUTES

ALL_MINING_MODEL_ATTRIBUTES describes the attributes of the mining models accessible to the current user.

Attributes correspond to the columns of data used to build a model. Columns with the same names and data types should be available in any data set to which the model is applied.

Related Views

- DBA_MINING_MODEL_ATTRIBUTES describes the attributes of all mining models in the database.
- USER_MINING_MODEL_ATTRIBUTES describes the attributes of the mining models owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the mining model
MODEL_NAME	VARCHAR2(30)	NOT NULL	Name of the mining model
ATTRIBUTE_NAME	VARCHAR2(30)	NOT NULL	Name of the attribute
ATTRIBUTE_TYPE	VARCHAR2(11)		Logical type of the attribute. The type is identified during the model build or apply process: ■ NUMERICAL - Numeric data ■ CATEGORICAL - Character data
DATA_TYPE	VARCHAR2(12)		Data type of the attribute: ■ VARCHAR2 ■ NUMBER ■ FLOAT ■ CHAR ■ NESTED TABLE
DATA_LENGTH	NUMBER		Length of the data type
DATA_PRECISION	NUMBER		Precision of a fixed point number. Precision, which is the total number of significant decimal digits, is represented as *p* in the data type NUMBER(p,s).
DATA_SCALE	NUMBER		Scale of a fixed point number. Scale, which is the number of digits from the decimal to the least significant digit, is represented as *s* in the data type NUMBER(p,s).
USAGE_TYPE	VARCHAR2(8)		Indicates whether the attribute was used to construct the model (ACTIVE) or not (INACTIVE). Some attributes may be eliminated by transformations or algorithmic processing. The *_MINING_MODEL_ATTRIBUTES view only lists the attributes used by the model, therefore the value of this column is always ACTIVE.
TARGET	VARCHAR2(3)		Indicates whether the attribute is the target of a predictive model (YES) or not (NO). The target describes the result that is produced when the model is applied.

See Also:

- "DBA_MINING_MODEL_ATTRIBUTES" on page 4-150
- "USER_MINING_MODEL_ATTRIBUTES" on page 5-116
- *Oracle Data Mining Concepts* for information about model attributes

ALL_MINING_MODEL_SETTINGS

ALL_MINING_MODEL_SETTINGS describes the settings of the mining models accessible to the current user.

Model settings can be specified in a settings table that is used as input to the model build process. The settings associated with a model are algorithm-dependent. Setting values can be provided as input to the model build process. If no values are provided, then default values are used.

Related Views

- DBA_MINING_MODEL_SETTINGS describes the settings of all mining models in the database.
- USER_MINING_MODEL_SETTINGS describes the settings of the mining models owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the mining model
MODEL_NAME	VARCHAR2(30)	NOT NULL	Name of the mining model
SETTING_NAME	VARCHAR2(30)	NOT NULL	Name of the setting
SETTING_VALUE	VARCHAR2(4000)		Value of the setting
SETTING_TYPE	VARCHAR2(7)		Indicates whether the default value (DEFAULT) or a user-specified value (INPUT) is used by the model

See Also:

- "DBA_MINING_MODEL_SETTINGS" on page 4-150
- "USER_MINING_MODEL_SETTINGS" on page 5-116
- *Oracle Database PL/SQL Packages and Types Reference* and *Oracle Data Mining Application Developer's Guide* for descriptions of model settings

ALL_MINING_MODELS

ALL_MINING_MODELS describes the mining models accessible to the current user.

Related Views

- DBA_MINING_MODELS describes all mining models in the database.
- USER_MINING_MODELS describes the mining models owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the mining model

Column	Datatype	NULL	Description
MODEL_NAME	VARCHAR2(30)	NOT NULL	Name of the mining model
MINING_FUNCTION	VARCHAR2(30)		Function of the mining model. The function identifies the class of problems that can be solved by this model. The mining function is specified when the model is built: ■ CLASSIFICATION ■ REGRESSION ■ CLUSTERING ■ FEATURE_EXTRACTION ■ ASSOCIATION_RULES ■ ATTRIBUTE_IMPORTANCE
ALGORITHM	VARCHAR2(30)		Algorithm used by the model. Each mining function has a default algorithm. The default can be overridden with a model setting (see *_MINING_MODEL_SETTINGS): ■ NAIVE_BAYES ■ ADAPTIVE_BAYES_NETWORK ■ DECISION_TREE ■ SUPPORT_VECTOR_MACHINES ■ KMEANS ■ O_CLUSTER ■ NONNEGATIVE_MATRIX_FACTOR ■ GENERALIZED_LINEAR_MODEL ■ APRIORI_ASSOCIATION_RULES ■ MINIMUM_DESCRIPTION_LENGTH
CREATION_DATE	DATE	NOT NULL	Date that the model was created
BUILD_DURATION	NUMBER		Time (in seconds) of the model build process
MODEL_SIZE	NUMBER		Size of the model (in megabytes)
COMMENTS	VARCHAR2(4000)		Comment applied to the model with a SQL COMMENT statement

See Also:

- "DBA_MINING_MODELS" on page 4-150
- "USER_MINING_MODELS" on page 5-116
- *Oracle Data Mining Administrator's Guide* for information about mining model schema objects
- *Oracle Data Mining Concepts* for information about Data Mining

ALL_MVIEW_AGGREGATES

ALL_MVIEW_AGGREGATES describes the grouping functions (aggregate operations) that appear in the SELECT list of materialized aggregate views accessible to the current user.

Related Views

- DBA_MVIEW_AGGREGATES describes all such grouping functions defined for all materialized views in the database.
- USER_MVIEW_AGGREGATES describes all such grouping functions defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that include references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this aggregation within the SELECT list. For the position of nonaggregate elements of the select list, see "ALL_MVIEW_KEYS" on page 2-92.
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of this column in the container table
AGG_FUNCTION	VARCHAR2(8)		Aggregation function
DISTINCTFLAG	VARCHAR2(1)		Whether this aggregation is distinct (Y | N)
MEASURE	LONG		SQL text of the measure, excluding the aggregation function. Equal to * for COUNT(*).

ALL_MVIEW_ANALYSIS

ALL_MVIEW_ANALYSIS describes the materialized views accessible to the current user. It provides additional information for analysis by applications. Minimal information is displayed for materialized views that do not support query rewrite (such as materialized views with remote master tables or nondeterministic functions).

Related Views

- DBA_MVIEW_ANALYSIS describes all such materialized views in the database.
- USER_MVIEW_ANALYSIS describes all such materialized views owned by the current user.

Note: All of the information in these views is also displayed in ALL_MVIEWS and its related views. Oracle recommends that you refer to ALL_MVIEWS for this information instead of these views.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MVIEW_TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the container table (see next column)
CONTAINER_NAME	VARCHAR2(30)		Name of the internal container in which the materialized view data is held. Normally this is the same as MVIEW_NAME. For materialized views created prior to Oracle8*i*, the Oracle Database attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, the Oracle Database truncates the name to 19 bytes and adds a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
LAST_REFRESH_SCN	NUMBER		System change number (SCN) of the last refresh operation
LAST_REFRESH_DATE	DATE		SYSDATE of the last refresh
REFRESH_METHOD	VARCHAR2(8)		Default refresh method: FORCE, FAST, COMPLETE, or NONE

Column	Datatype	NULL	Description
SUMMARY	VARCHAR2(1)		Whether this materialized view includes a GROUP BY clause or aggregation (Y | N)
FULLREFRESHTIM	NUMBER		Approximate refresh time, in seconds, for full refresh (defined only when SUMMARY = Y)
INCREFRESHTIM	NUMBER		Approximate refresh time, in seconds, for fast refresh (defined only when SUMMARY = Y)
CONTAINS_VIEWS	VARCHAR2(1)		Whether this materialized view contains a view in its definition (Y | N)
UNUSABLE	VARCHAR2(1)		Whether this materialized view is UNUSABLE (inconsistent data) (Y | N). A materialized view can be UNUSABLE if a system failure occurs during a full refresh.
RESTRICTED_SYNTAX	VARCHAR2(1)		Whether this materialized view had a restriction in its defining query that limits the use of query rewrite (Y | N). More complete information is provided by the REWRITE_CAPABILITY column of the ALL_, DBA_, and USER_MVIEWS views.
INC_REFRESHABLE	VARCHAR2(1)		Whether this materialized view can be fast refreshed (Y | N)
KNOWN_STALE	VARCHAR2(1)		Whether the data contained in the materialized view is known to be inconsistent with the master table data because that has been updated since the last successful refresh (Y | N)
INVALID	VARCHAR2(1)		Whether this materialized view is in an invalid state (inconsistent metadata) (Y | N)
REWRITE_ENABLED	VARCHAR2(1)		Whether this materialized view is currently enabled for query rewrite (Y | N)
QUERY_LEN	NUMBER		Length (in bytes) of the query field
QUERY	LONG		SELECT expression of the materialized view definition
REVISION	NUMBER		Reserved for internal use

ALL_MVIEW_COMMENTS

ALL_MVIEW_COMMENTS displays comments on the materialized views accessible to the current user.

Related Views

- DBA_MVIEW_COMMENTS displays comments on the materialized views in the database.
- USER_MVIEW_COMMENTS displays comments on the materialized views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
COMMENTS	VARCHAR2(4000)		Comment on the materialized view

See Also:

- "DBA_MVIEW_COMMENTS" on page 4-150
- "USER_MVIEW_COMMENTS" on page 5-117

ALL_MVIEW_DETAIL_PARTITION

ALL_MVIEW_DETAIL_PARTITION displays the freshness information of the materialized views, with respect to a PCT detail partition, accessible to the current user.

Related Views

- DBA_MVIEW_DETAIL_PARTITION displays freshness information for all materialized views in the database, with respect to a PCT detail partition.
- USER_MVIEW_DETAIL_PARTITION displays freshness information for all materialized views, with respect to a PCT detail partition, owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialzed view
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the detail object
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Name of the detail object
DETAIL_PARTITION_NAME	VARCHAR2(30)		Name of the detail object partition
DETAIL_PARTITION_POSITION	NUMBER		Position of the detail object partition
FRESHNESS	CHAR(5)		Freshness state (FRESH, STALE, UNKNOWN, NA, and so on)

ALL_MVIEW_DETAIL_RELATIONS

ALL_MVIEW_DETAIL_RELATIONS describes the named detail relations that are either specified in the FROM list of the subquery that defines a materialized view accessible to the current user, or that are indirectly referenced through views in that FROM list. Inline views in the materialized view definition are not represented in this view or the related views.

Related Views

- DBA_MVIEW_DETAIL_RELATIONS describes all such detail relations defined for all materialized views in the database.
- USER_MVIEW_DETAIL_RELATIONS describes such detail relations defined for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as *snapshots* prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (that is, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(9)		TABLE, VIEW, SNAPSHOT, CONTAINER, or UNDEFINED
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation

Column	Datatype	NULL	Description
DETAILOBJ_PCT	VARCHAR2(1)		Is the detail object PCT supported? (Y or N)
NUM_FRESH_PCT_PARTITIONS	NUMBER		Number of fresh PCT partitions
NUM_STALE_PCT_PARTITIONS	NUMBER		Number of stale PCT partitions

ALL_MVIEW_DETAIL_SUBPARTITION

ALL_MVIEW_DETAIL_SUBPARTITION displays the freshness information of the materialized views, with respect to a PCT detail subpartition, accessible to the current user.

Related Views

- DBA_MVIEW_DETAIL_SUBPARTITION displays freshness information for all materialized views in the database, with respect to a PCT detail subpartition.
- USER_MVIEW_DETAIL_SUBPARTITION displays freshness information for all materialized views, with respect to a PCT detail subpartition, owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the detail object
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Name of the detail object
DETAIL_PARTITION_NAME	VARCHAR2(30)		Name of the detail object partition
DETAIL_SUBPARTITION_NAME	VARCHAR2(30)		Name of the detail object subpartition
DETAIL_SUBPARTITION_POSITION	NUMBER		Position of the detail object subpartition
FRESHNESS	CHAR(5)		Freshness state (FRESH, STALE, UNKNOWN, NA, and so on)

ALL_MVIEW_JOINS

ALL_MVIEW_JOINS describes joins between two columns in the WHERE clause of the subquery that defines a materialized view accessible to the current user.

Related Views

- DBA_MVIEW_JOINS describes all such joins for all materialized views in the database.
- USER_MVIEW_JOINS describes such joins for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as "snapshots" prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view

Column	Datatype	NULL	Description
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
DETAILOBJ1_OWNER	VARCHAR2(30)	NOT NULL	Owner of the first object in the join[1]
DETAILOBJ1_RELATION	VARCHAR2(30)	NOT NULL	Name of the first object in the join[1]
DETAILOBJ1_COLUMN	VARCHAR2(30)	NOT NULL	Join column of the first object in the join[1]
OPERATOR	CHAR(1)		Join operator[1]
OPERATOR_TYPE	VARCHAR2(1)		Whether the join is an inner or outer join[1]
DETAILOBJ2_OWNER	VARCHAR2(30)	NOT NULL	Owner of the second object in the join[1]
DETAILOBJ2_RELATION	VARCHAR2(30)	NOT NULL	Name of the second object in the join[1]
DETAILOBJ2_COLUMN	VARCHAR2(30)	NOT NULL	Join column of the second object in the join[1]

[1] These rows relate only to materialized join views and materialized aggregate views. They describe the two detail objects of a materialized view join.

ALL_MVIEW_KEYS

ALL_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views accessible to the current user are based.

Related Views

- DBA_MVIEW_KEYS describes such columns and expressions for all materialized views in the database.
- USER_MVIEW_KEYS describes such columns and expressions for all materialized views owned by the current user.

All three views exclude materialized views that reference remote tables or that includes references to a nonstatic value such as SYSDATE or USER. These views also exclude materialized views that were created as *snapshots* prior to Oracle8*i* and that were never altered to enable query rewrite.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Materialized view name
POSITION_IN_SELECT	NUMBER	NOT NULL	Ordinal position of this key within the SELECT list
CONTAINER_COLUMN	VARCHAR2(30)	NOT NULL	Name of the column in the container table
DETAILOBJ_OWNER	VARCHAR2(30)	NOT NULL	Detail object owner
DETAILOBJ_NAME	VARCHAR2(30)	NOT NULL	Detail object name (for example, the name of a table or view)
DETAILOBJ_TYPE	VARCHAR2(5)		Detail object type (VIEW | TABLE)
DETAILOBJ_ALIAS	VARCHAR2(30)		Implicit or explicit alias for detail relation
DETAILOBJ_COLUMN	VARCHAR2(30)	NOT NULL	Name of the detail relation column

ALL_MVIEW_LOGS

ALL_MVIEW_LOGS describes all materialized view logs accessible to the current user.

Related Views

- DBA_MVIEW_LOGS describes all materialized view logs in the database.

- USER_MVIEW_LOGS describes all materialized view logs owned by the current user.

Column	Datatype	NULL	Description
LOG_OWNER	VARCHAR2(30)		Owner of the materialized view log
MASTER	VARCHAR2(30)		Name of the master table or master materialized view whose changes are logged
LOG_TABLE	VARCHAR2(30)		Name of the table where the changes to the master table or master materialized view are logged
LOG_TRIGGER	VARCHAR2(30)		Obsolete with the release of Oracle8*i* and higher. Set to NULL. Formerly, this parameter was an after-row trigger on the master which inserted rows into the log.
ROWIDS	VARCHAR2(3)		If YES, records rowid information
PRIMARY_KEY	VARCHAR2(3)		If YES, records primary key information
OBJECT_ID	VARCHAR2(3)		If YES, records object identifier information in an object table
FILTER_COLUMNS	VARCHAR2(3)		If YES, records filter columns
SEQUENCE	VARCHAR2(3)		If YES, records the sequence value, which provides additional ordering information
INCLUDE_NEW_VALUES	VARCHAR2(3)		If YES, records both old and new values. If NO, records old values, but does not record new values.

See Also:

- "DBA_MVIEW_LOGS" on page 4-151
- "USER_MVIEW_LOGS" on page 5-117

ALL_MVIEW_REFRESH_TIMES

ALL_MVIEW_REFRESH_TIMES describes refresh times of the materialized views accessible to the current user.

Related Views

- DBA_MVIEW_REFRESH_TIMES describes refresh times of all materialized views in the database.
- USER_MVIEW_REFRESH_TIMES describes refresh times of the materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MASTER_OWNER	VARCHAR2(30)		Owner of the master table
MASTER	VARCHAR2(30)		Name of the master table
LAST_REFRESH	DATE		The last refresh

See Also:

- "DBA_MVIEW_REFRESH_TIMES" on page 4-152
- "USER_MVIEW_REFRESH_TIMES" on page 5-118

ALL_MVIEWS

ALL_MVIEWS describes all materialized views accessible to the current user.

Related views

- DBA_MVIEWS describes all materialized views in the database.
- USER_MVIEWS describes all materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Schema in which the materialized view was created
MVIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
CONTAINER_NAME	VARCHAR2(30)	NOT NULL	Name of the container in which the materialized view's data is held. Normally this is the same as MVIEW_NAME. For materialized views created prior to Oracle8*i*, the Oracle Database attaches the 6-byte prefix SNAP$_. If MVIEW_NAME has more than 19 bytes, then the Oracle Database truncates the name to 19 bytes and may add a 4-byte sequence number as a suffix to produce a nonambiguous CONTAINER_NAME.
QUERY	LONG		Query that defines the materialized view
QUERY_LEN	NUMBER(38)		Length (in bytes) of the defining query
UPDATABLE	VARCHAR2(1)		Indicates whether the materialized view is updatable (Y) or not (N)
UPDATE_LOG	VARCHAR2(30)		For updatable materialized views, the filename of the update log
MASTER_ROLLBACK_SEG	VARCHAR2(30)		Rollback segment for the master site or the master materialized view site
MASTER_LINK	VARCHAR2(128)		Database link for the master site or the master materialized view site
REWRITE_ENABLED	VARCHAR2(1)		Indicates whether rewrite is enabled (Y) or not (N)
REWRITE_CAPABILITY	VARCHAR2(9)		Indicates whether the materialized view is eligible for rewrite, and if so, what rules must be followed: ■ NONE - Materialized view cannot be used for rewrite, because rewrite is disallowed or prevented ■ TEXTMATCH - Defining query of the materialized view contained restrictions on the use of query rewrite ■ GENERAL - Defining query of the materialized view contained no restrictions on the use of query rewrite, so the Oracle Database can apply any rewrite rule that is supported
REFRESH_MODE	VARCHAR2(6)		Refresh mode of the materialized view: ■ DEMAND - Oracle Database refreshes this materialized view whenever an appropriate refresh procedure is called ■ COMMIT - Oracle Database refreshes this materialized view when a transaction on one of the materialized view's masters commits ■ NEVER - Oracle Database never refreshes this materialized view

Column	Datatype	NULL	Description
REFRESH_METHOD	VARCHAR2(8)		Default method used to refresh the materialized view (can be overridden through the API): ■ COMPLETE (C) - Materialized view is completely refreshed from the masters ■ FORCE (?) - Oracle Database performs a fast refresh if possible, otherwise a complete refresh ■ FAST (F) - Oracle Database performs an incremental refresh applying changes that correspond to changes in the masters since the last refresh ■ NEVER (N) - User specified that the Oracle Database should not refresh this materialized view
BUILD_MODE	VARCHAR2(9)		Indicates how the materialized view was populated during creation: ■ IMMEDIATE - Populated from the masters during creation ■ DEFERRED - Not populated during creation. Must be explicitly populated later by the user. ■ PREBUILT - Populated with an existing table during creation. The relationship of the contents of this prebuilt table to the materialized view's masters is unknown to the Oracle Database.
FAST_REFRESHABLE	VARCHAR2(18)		Indicates whether the materialized view is eligible for incremental (fast) refresh. The Oracle Database calculates this value statically, based on the materialized view definition query: ■ NO - Materialized view is not fast refreshable, and hence is complex ■ DIRLOAD - Fast refresh is supported only for direct loads ■ DML - Fast refresh is supported only for DML operations ■ DIRLOAD_DML - Fast refresh is supported for both direct loads and DML operations ■ DIRLOAD_LIMITEDDML - Fast refresh is supported for direct loads and a subset of DML operations
LAST_REFRESH_TYPE	VARCHAR2(8)		Method used for the most recent refresh: ■ COMPLETE - Most recent refresh was complete ■ FAST - Most recent refresh was fast (incremental) ■ NA - Materialized view has not yet been refreshed (for example, if it was created DEFERRED)
LAST_REFRESH_DATE	DATE		Date on which the materialized view was most recently refreshed. Blank if not yet populated.

Column	Datatype	NULL	Description
STALENESS	VARCHAR2(19)		Relationship between the contents of the materialized view and the contents of the materialized view's masters: ■ FRESH - Materialized view is a read-consistent view of the current state of its masters ■ STALE - Materialized view is out of date because one or more of its masters has changed. If the materialized view was FRESH before it became STALE, then it is a read-consistent view of a former state of its masters. ■ UNUSABLE - Materialized view is not a read-consistent view of its masters from any point in time ■ UNKNOWN - Oracle Database does not know whether the materialized view is in a read-consistent view of its masters from any point in time (this is the case for materialized views created on prebuilt tables) ■ UNDEFINED - Materialized view has remote masters. The concept of staleness is not defined for such materialized views.
AFTER_FAST_REFRESH	VARCHAR2(19)		Specifies the staleness value that will occur if a fast refresh is applied to this materialized view. Its values are the same as for the STALENESS column, plus the value NA, which is used when fast refresh is not applicable to this materialized view.
UNKNOWN_PREBUILT	VARCHAR2(1)		Indicates whether the materialized view is prebuilt (Y) or not (N)
UNKNOWN_PLSQL_FUNC	VARCHAR2(1)		Indicates whether the materialized view contains PL/SQL functions (Y) or not (N)
UNKNOWN_EXTERNAL_TABLE	VARCHAR2(1)		Indicates whether the materialized view contains external tables (Y) or not (N)
UNKNOWN_CONSIDER_FRESH	VARCHAR2(1)		Indicates whether the materialized view is considered fresh (Y) or not (N)
UNKNOWN_IMPORT	VARCHAR2(1)		Indicates whether the materialized view is imported (Y) or not (N)
UNKNOWN_TRUSTED_FD	VARCHAR2(1)		Indicates whether the materialized view uses trusted constraints for refresh (Y) or not (N)
COMPILE_STATE	VARCHAR2(19)		Validity of the materialized view with respect to the objects upon which it depends: ■ VALID - Materialized view has been validated without error, and no object upon which it depends has changed since the last validation ■ NEEDS_COMPILE - Some object upon which the materialized view depends has changed. An ALTER MATERIALIZED VIEW...COMPILE statement is required to validate this materialized view. ■ ERROR - Materialized view has been validated with one or more errors
USE_NO_INDEX	VARCHAR2(1)		Indicates whether the materialized view was created using the USING NO INDEX clause (Y) or the materialized view was created with the default index (N). The USING NO INDEX clause suppresses the creation of the default index.
STALE_SINCE	DATE		Time from when the materialized view became stale
NUM_PCT_TABLES	NUMBER		Number of PCT detail tables
NUM_FRESH_PCT_REGIONS	NUMBER		Number of fresh PCT partition regions
NUM_STALE_PCT_REGIONS	NUMBER		Number of stale PCT partition regions

See Also:

- "DBA_MVIEWS" on page 4-152
- "USER_MVIEWS" on page 5-118
- *Oracle Database Advanced Replication* for more information on materialized views to support replication
- *Oracle Database Data Warehousing Guide* for more information on materialized views to support data warehousing

ALL_NESTED_TABLE_COLS

ALL_NESTED_TABLE_COLS describes the columns of the nested tables accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

Related Views

- DBA_NESTED_TABLE_COLS describes the columns of all nested tables in the database.
- USER_NESTED_TABLE_COLS describes the columns of the nested tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the nested table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the nested table
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER		Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column[1]
LOW_VALUE	RAW(32)		Low value in the column[1]
HIGH_VALUE	RAW(32)		High value in the column[1]
DENSITY	NUMBER		Density of the column[1]
NUM_NULLS	NUMBER		Number of nulls in the column

Column	Datatype	NULL	Description
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column **Note:** The number of buckets in a histogram is specified in the SIZE parameter of the SQL statement ANALYZE. However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2(1)		B | C. B indicates that the column uses BYTE length semantics. C indicates that the column uses CHAR length semantics. NULL indicates the datatype is not any of the following: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2(1)		Indicates whether the column data is in release 8.0 image format (Y) or not (N)
DATA_UPGRADED	VARCHAR2(1)		Indicates whether the column data has been upgraded to the latest type version format (Y) or not (N)
HIDDEN_COLUMN	VARCHAR2(3)		Indicates whether the column is a hidden column (YES) or not (NO)
VIRTUAL_COLUMN	VARCHAR2(3)		Indicates whether the column is a virtual column (YES) or not (NO)
SEGMENT_COLUMN_ID	NUMBER		Sequence number of the column in the segment
INTERNAL_COLUMN_ID	NUMBER	NOT NULL	Internal sequence number of the column
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED
QUALIFIED_COL_NAME	VARCHAR2(4000)		Qualified column name

[1] These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views.

ALL_NESTED_TABLES

ALL_NESTED_TABLES describes the nested tables in tables accessible to the current user.

Related Views

- DBA_NESTED_TABLES describes all nested tables in the database.
- USER_NESTED_TABLES describes nested tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the nested table
TABLE_NAME	VARCHAR2(30)		Name of the nested table
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of which the nested table was created
TABLE_TYPE_NAME	VARCHAR2(30)		Name of the type of the nested table
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the parent table containing the nested table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Column name of the parent table that corresponds to the nested table
STORAGE_SPEC	VARCHAR2(30)		Whether storage for the nested table is USER-SPECIFIED or DEFAULT
RETURN_TYPE	VARCHAR2(20)		Return type of the varray column (LOCATOR | VALUE)
ELEMENT_SUBSTITUTABLE	VARCHAR2(25)		Indicates whether the nested table element is substitutable (Y) or not (N)

ALL_OBJ_COLATTRS

ALL_OBJ_COLATTRS describes object columns and attributes contained in the tables accessible to the current user.

Related Views

- DBA_OBJ_COLATTRS describes object columns and attributes contained in all tables in the database.
- USER_OBJ_COLATTRS describes object columns and attributes contained in the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table containing the object column or attribute
COLUMN_NAME	VARCHAR2(4000)		Fully qualified name of the object column or attribute
SUBSTITUTABLE	VARCHAR2(15)		Indicates whether the column is substitutable (Y) or not (N)

See Also:

- "DBA_OBJ_COLATTRS" on page 5-3
- "USER_OBJ_COLATTRS" on page 5-119

ALL_OBJECT_TABLES

ALL_OBJECT_TABLES describes the object tables accessible to the current user.

Related Views

- DBA_OBJECT_TABLES describes all object tables in the database.
- USER_OBJECT_TABLES describes the object tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; NULL for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
STATUS	VARCHAR2(8)		If a previous DROP TABLE operation failed, indicates whether the table is unusable (UNUSABLE) or valid (VALID)
PCT_FREE	NUMBER		Minimum percentage of free space in a block; NULL for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; NULL for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; NULL for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; NULL for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); NULL for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); NULL for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; NULL for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; NULL for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; NULL for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; NULL for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment; NULL for partitioned tables
LOGGING	VARCHAR2(3)		Logging attribute
BACKED_UP	VARCHAR2(1)		Whether the table has been backed up since last modification
NUM_ROWS	NUMBER		Number of rows in the table
BLOCKS	NUMBER		Number of used blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the table
AVG_SPACE	NUMBER		Average available free space in the table
CHAIN_CNT	NUMBER		Number of chained rows in the table

Column	Datatype	NULL	Description
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average free space of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of parallel execution processes per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)
TABLE_LOCK	VARCHAR2(8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Whether the table is partitioned (YES | NO)
IOT_TYPE	VARCHAR2(12)		If index-organized table, then IOT_TYPE is IOT or IOT_OVERFLOW else NULL
OBJECT_ID_TYPE	VARCHAR2(16)		Whether the object ID (OID) is USER-DEFINED or SYSTEM-GENERATED
TABLE_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the table
TABLE_TYPE	VARCHAR2(30)		Type of the table
TEMPORARY	VARCHAR2(1)		Whether this is a temporary table
SECONDARY	VARCHAR2(1)		Whether the object table is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y | N)
NESTED	VARCHAR2(3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used for table blocks
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is ENABLED or DISABLED
GLOBAL_STATS	VARCHAR2(3)		For partitioned object tables, indicates whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION: the rows are preserved for the duration of the session ■ SYS$TRANSACTION: the rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2(8)		Whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set
CLUSTER_OWNER	VARCHAR2(30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2(8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); NULL for partitioned tables

Column	Datatype	NULL	Description
DROPPED	VARCHAR2(3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); NULL for partitioned tables

See Also:

- "DBA_OBJECT_TABLES" on page 5-3
- "USER_OBJECT_TABLES" on page 5-119

ALL_OBJECTS

ALL_OBJECTS describes all objects accessible to the current user.

Related Views

- DBA_OBJECTS describes all objects in the database.
- USER_OBJECTS describes all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject (for example, partition)
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object
DATA_OBJECT_ID	NUMBER		Dictionary object number of the segment that contains the object **Note:** OBJECT_ID and DATA_OBJECT_ID display data dictionary metadata. Do not confuse these numbers with the unique 16-byte object identifier (*object ID*) that the Oracle Database assigns to row objects in object tables in the system.
OBJECT_TYPE	VARCHAR2(19)		Type of the object (such as TABLE, INDEX)
CREATED	DATE	NOT NULL	Timestamp for the creation of the object
LAST_DDL_TIME	DATE	NOT NULL	Timestamp for the last modification of the object resulting from a DDL statement (including grants and revokes)
TIMESTAMP	VARCHAR2(20)		Timestamp for the specification of the object (character data)
STATUS	VARCHAR2(7)		Status of the object (VALID, INVALID, or N/A)
TEMPORARY	VARCHAR2(1)		Whether the object is temporary (the current session can see only data that it placed in this object itself)
GENERATED	VARCHAR2(1)		Indicates whether the name of this object was system generated (Y) or not (N)
SECONDARY	VARCHAR2(1)		Whether this is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y | N)
NAMESPACE	NUMBER		Namespace for the object
EDITION_NAME	VARCHAR2(30)		Name of the Application Edition where the object is identified

ALL_OPANCILLARY

ALL_OPANCILLARY describes operators whose bindings are ancillary to other (primary) operators.

Related Views

- DBA_OPANCILLARY provides such information about all operators in the database.
- USER_OPANCILLARY provides such information about operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the ancillary operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the ancillary operator
BINDING#	NUMBER	NOT NULL	Binding number the of ancillary operator
PRIMOP_OWNER	VARCHAR2(30)	NOT NULL	Owner of the primary operator
PRIMOP_NAME	VARCHAR2(30)	NOT NULL	Name of the primary operator
PRIMOP_BIND#	NUMBER	NOT NULL	Binding number of the primary operator

ALL_OPARGUMENTS

ALL_OPARGUMENTS describes arguments for each operator binding accessible to the current user.

Related Views

- DBA_OPARGUMENTS describes arguments of all operator bindings in the database.
- USER_OPARGUMENTS describes arguments of all operator bindings owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator argument
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator argument
BINDING#	NUMBER	NOT NULL	Binding number of the operator argument
POSITION	NUMBER	NOT NULL	Position of the operator argument (1, 2, 3, ...)
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the operator argument

ALL_OPBINDINGS

ALL_OPBINDINGS describes the binding functions and methods on the operators accessible to the current user.

Related Views

- DBA_OPBINDINGS describes the binding functions and methods on all operators in the database.
- USER_OPBINDINGS describes the binding functions and methods on the operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
BINDING#	NUMBER	NOT NULL	Binding number of the operator
FUNCTION_NAME	VARCHAR2(92)		Name of the binding function or method as specified by the user
RETURN_SCHEMA	VARCHAR2(30)		Name of the schema of the return type if the return type of the binding is an object type
RETURN_TYPE	VARCHAR2(30)		Name of the return type
IMPLEMENTATION_TYPE_SCHEMA	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT or SCAN CONTEXT, then this column displays the schema of the implementation type used by the functional implementation of the operator as a scan context (null if the operator was created without this syntax). **See Also:** the CREATE OPERATOR statement in *Oracle Database SQL Language Reference*
IMPLEMENTATION_TYPE	VARCHAR2(30)		If the operator was created WITH INDEX CONTEXT or SCAN CONTEXT, then this column displays the name of the implementation type used by the functional implementation of the operator as a scan context (null if the operator was created without this syntax). **See Also:** the CREATE OPERATOR statement in *Oracle Database SQL Language Reference*
PROPERTY	VARCHAR2(43)		Property of the operator binding: ■ WITH INDEX CONTEXT ■ COMPUTE ANCILLARY DATA ■ ANCILLARY TO ■ WITH COLUMN CONTEXT ■ WITH INDEX, COLUMN CONTEXT ■ COMPUTE ANCILLARY DATA, WITH COLUMN CONTEXT

See Also:

- "DBA_OPBINDINGS" on page 5-4
- "USER_OPBINDINGS" on page 5-119

ALL_OPERATOR_COMMENTS

ALL_OPERATOR_COMMENTS displays comments for the user-defined operators accessible to the current user.

Related Views

- DBA_OPERATOR_COMMENTS displays comments for all user-defined operators in the database.
- USER_OPERATOR_COMMENTS displays comments for the user-defined operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the user-defined operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the user-defined operator
COMMENTS	VARCHAR2(4000)		Comment for the user-defined operator

See Also:

- "DBA_OPERATOR_COMMENTS" on page 5-4
- "USER_OPERATOR_COMMENTS" on page 5-119

ALL_OPERATORS

ALL_OPERATORS describes operators accessible to the current user.

Related Views

- DBA_OPERATORS describes all operators in the database.
- USER_OPERATORS describes all operators owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the operator
OPERATOR_NAME	VARCHAR2(30)	NOT NULL	Name of the operator
NUMBER_OF_BINDS	NUMBER	NOT NULL	Number of bindings associated with the operator

See Also:

- "DBA_OPERATORS" on page 5-4
- "USER_OPERATORS" on page 5-120

ALL_OUTLINE_HINTS

ALL_OUTLINE_HINTS is a synonym for USER_OUTLINE_HINTS.

See Also: "USER_OUTLINE_HINTS" on page 5-120

ALL_OUTLINES

ALL_OUTLINES is a synonym for USER_OUTLINES.

See Also: "USER_OUTLINES" on page 5-120

3

Static Data Dictionary Views: ALL_PART_COL_STATISTICS to DATABASE_PROPERTIES

This chapter contains the static data dictionary views ALL_NESTED_TABLE_COLS through DATABASE_PROPERTIES.

ALL_PART_COL_STATISTICS

ALL_PART_COL_STATISTICS displays column statistics and histogram information for the table partitions accessible to the current user.

Related Views

- DBA_PART_COL_STATISTICS displays column statistics and histogram information for all table partitions in the database.
- USER_PART_COL_STATISTICS displays column statistics and histogram information for the table partitions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(4000)		Name of the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the column
LAST_ANALYZED	DATE		Date on which the column was most recently analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

Column	Datatype	NULL	Description
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: - NONE - FREQUENCY - HEIGHT BALANCED

See Also:

- "DBA_PART_COL_STATISTICS" on page 5-7
- "USER_PART_COL_STATISTICS" on page 5-120

ALL_PART_HISTOGRAMS

ALL_PART_HISTOGRAMS displays the histogram data (endpoints per histogram) for the histograms on the table partitions accessible to the current user.

Related Views

- DBA_PART_HISTOGRAMS displays the histogram data for the histograms on all table partitions in the database.
- USER_PART_HISTOGRAMS displays the histogram data for the histograms on the table partitions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the table partition
COLUMN_NAME	VARCHAR2(4000)		Name of the column
BUCKET_NUMBER	NUMBER		Bucket number of the histogram
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for the bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for the bucket

Note: These views are populated only if you collect statistics on the index using the ANALYZE statement or the DBMS_STATS package.

See Also:

- "DBA_PART_HISTOGRAMS" on page 5-7
- "USER_PART_HISTOGRAMS" on page 5-120

ALL_PART_INDEXES

ALL_PART_INDEXES displays the object-level partitioning information for the partitioned indexes accessible to the current user.

Related Views

- DBA_PART_INDEXES displays the object-level partitioning information for all partitioned indexes in the database.
- USER_PART_INDEXES displays the object-level partitioning information for the partitioned indexes owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table
PARTITIONING_TYPE	VARCHAR2(7)		Type of the partitioning method: ■ RANGE ■ HASH ■ SYSTEM ■ LIST
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of the composite partitioning method: ■ NONE ■ RANGE ■ HASH ■ SYSTEM ■ LIST
PARTITION_COUNT	NUMBER	NOT NULL	Number of partitions in the index
DEF_SUBPARTITION_COUNT	NUMBER		For a composite-partitioned index, the default number of subpartitions, if specified
PARTITIONING_KEY_COUNT	NUMBER	NOT NULL	Number of columns in the partitioning key
SUBPARTITIONING_KEY_COUNT	NUMBER		For a composite-partitioned index, the number of columns in the subpartitioning key
LOCALITY	VARCHAR2(6)		Indicates whether the partitioned index is local (LOCAL) or global (GLOBAL)
ALIGNMENT	VARCHAR2(12)		Indicates whether the partitioned index is prefixed (PREFIXED) or non-prefixed (NON_PREFIXED)
DEF_TABLESPACE_NAME	VARCHAR2(30)		For a local index, the default tablespace to be used when adding or splitting a table partition
DEF_PCT_FREE	NUMBER	NOT NULL	For a local index, the default PCTFREE value to be used when adding a table partition
DEF_INI_TRANS	NUMBER	NOT NULL	For a local index, the default INITRANS value to be used when adding a table partition
DEF_MAX_TRANS	NUMBER	NOT NULL	For a local index, the default MAXTRANS value to be used when adding a table partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		For a local index, the default INITIAL value (in Oracle blocks) to be used when adding a table partition, or DEFAULT if no INITIAL value was specified
DEF_NEXT_EXTENT	VARCHAR2(40)		For a local index, the default NEXT value (in Oracle blocks) to be used when adding a table partition, or DEFAULT if no NEXT value was specified
DEF_MIN_EXTENTS	VARCHAR2(40)		For a local index, the default MINEXTENTS value to be used when adding a table partition, or DEFAULT if no MINEXTENTS value was specified
DEF_MAX_EXTENTS	VARCHAR2(40)		For a local index, the default MAXEXTENTS value to be used when adding a table partition, or DEFAULT if no MAXEXTENTS value was specified

Column	Datatype	NULL	Description
DEF_MAX_SIZE	VARCHAR2(40)		For a local index, the default MAXSIZE value to be used when adding a table partition, or DEFAULT if no MAXSIZE value was specified
DEF_PCT_INCREASE	VARCHAR2(40)		For a local index, the default PCTINCREASE value to be used when adding a table partition, or DEFAULT if no PCTINCREASE value was specified
DEF_FREELISTS	NUMBER	NOT NULL	For a local index, the default FREELISTS value to be used when adding a table partition
DEF_FREELIST_GROUPS	NUMBER	NOT NULL	For a local index, the default FREELIST GROUPS value to be used when adding a table partition
DEF_LOGGING	VARCHAR2(7)		For a local index, the default LOGGING attribute to be used when adding a table partition: ■ NONE ■ YES ■ NO ■ UNKNOWN
DEF_BUFFER_POOL	VARCHAR2(7)		For a local index, the default buffer pool to be used when adding a table partition: ■ DEFAULT ■ KEEP ■ RECYCLE ■ NULL
DEF_PARAMETERS	VARCHAR2(1000)		Default parameter string for domain indexes
INTERVAL	VARCHAR2(1000)		String of the interval value

See Also:

- "DBA_PART_INDEXES" on page 5-7
- "USER_PART_INDEXES" on page 5-120

ALL_PART_KEY_COLUMNS

ALL_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects accessible to the current user.

Related Views

- DBA_PART_KEY_COLUMNS describes the partitioning key columns for all partitioned objects in the database.
- USER_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the partitioned table or index
NAME	VARCHAR2(30)		Name of the partitioned table or index
OBJECT_TYPE	CHAR(5)		Object type: ■ TABLE ■ INDEX
COLUMN_NAME	VARCHAR2(4000)		Name of the column
COLUMN_POSITION	NUMBER		Position of the column within the partitioning key

See Also:

- "DBA_PART_KEY_COLUMNS" on page 5-7
- "USER_PART_KEY_COLUMNS" on page 5-120

ALL_PART_LOBS

ALL_PART_LOBS displays table-level information about the partitioned LOBs accessible to the current user, including default attributes for LOB data partitions.

Related Views

- DBA_PART_LOBS displays table-level information about all partitioned LOBs in the database.
- USER_PART_LOBS displays table-level information about the partitioned LOBs owned by the current user. This view does not display the TABLE_OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the partitioned table containing the LOBs
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned table containing the LOBs
COLUMN_NAME	VARCHAR2(4000)		Name of the LOB column
LOB_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB
LOB_INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the partitioned LOB index
DEF_CHUNK	NUMBER	NOT NULL	Default value of CHUNK for a LOB data partition to be used when adding a partition
DEF_PCTVERSION	NUMBER	NOT NULL	Default value of PCTVERSION for a LOB data partition to be used when adding a partition
DEF_CACHE	VARCHAR2(10)		Indicates whether and how the LOB data is cached by default in the buffer cache: ■ YES - LOB data is placed in the buffer cache ■ NO - LOB data either is not brought into the buffer cache or is brought into the buffer cache and placed at the least recently used end of the LRU list ■ CACHEREADS - LOB data is brought into the buffer cache only during read operations but not during write operations
DEF_IN_ROW	VARCHAR2(3)		Indicates whether LOB data < 4000 bytes is stored by default inline (in the row) (YES) or not (NO); that is, whether or not ENABLE STORAGE IN ROW was specified when the LOB column was created or last altered
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default tablespace for a LOB data partition to be used when adding a partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL for a LOB data partition to be used when adding a partition, or DEFAULT if no INITIAL value was specified
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT for a LOB data partition to be used when adding a partition, or DEFAULT if no NEXT value was specified
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENTS for a LOB data partition to be used when adding a partition, or DEFAULT if no MINEXTENTS value was specified
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS for a LOB data partition to be used when adding a partition, or DEFAULT if no MAXEXTENTS value was specified

Column	Datatype	NULL	Description
DEF_MAX_SIZE	VARCHAR2(40)		Default value of MAXSIZE for a LOB data partition to be used when adding a partition, or DEFAULT if no MAXSIZE value was specified
DEF_RETENTION	VARCHAR2(7)		Default value of RETENTION for a LOB data partition to be used when adding a partition: ■ NONE ■ AUTO ■ MIN ■ MAX ■ DEFAULT ■ INVALID
DEF_MINRET	VARCHAR2(40)		Default value of RETENTION MIN for a LOB data partition to be used when adding a partition, or DEFAULT if no RETENTION MIN value was specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE for a LOB data partition to be used when adding a partition, or DEFAULT if no PCTINCREASE value was specified
DEF_FREELISTS	VARCHAR2(40)		Default value of FREELISTS for a LOB data partition to be used when adding a partition, or DEFAULT if no FREELISTS value was specified
DEF_FREELIST_GROUPS	VARCHAR2(40)		Default value of FREELIST GROUPS for a LOB data partition to be used when adding a partition, or DEFAULT if no FREELIST GROUPS value was specified
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute for a LOB data partition to be used when adding a partition: ■ NONE ■ YES ■ NO ■ UNKNOWN
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool for a LOB data partition to be used when adding a partition: ■ DEFAULT ■ KEEP ■ RECYCLE ■ NULL
DEF_ENCRYPT	VARCHAR2(4)		Default value of ENCRYPT for a LOB data partition to be used when adding a partition: ■ YES ■ NO ■ NONE
DEF_COMPRESS	VARCHAR2(6)		Default value of COMPRESS for a LOB data partition to be used when adding a partition: ■ LOW ■ MEDIUM ■ HIGH ■ NO ■ NONE

Column	Datatype	NULL	Description
DEF_DEDUPLICATE	VARCHAR2(15)		Default value of DEDUPLICATE for a LOB data partition to be used when adding a partition: - LOB - OBJECT - LOB VALIDATE - OBJECT VALIDATE - NO - NONE
DEF_SECUREFILE	VARCHAR2(3)		Indicates whether the LOB is a SecureFile LOB (YES) or not (NO)

See Also:

- "DBA_PART_LOBS" on page 5-7
- "USER_PART_LOBS" on page 5-121

ALL_PART_TABLES

ALL_PART_TABLES displays the object-level partitioning information for the partitioned tables accessible to the current user.

Related Views

- DBA_PART_TABLES displays the object-level partitioning information for all partitioned tables in the database.
- USER_PART_TABLES displays the object-level partitioning information for the partitioned tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the partitioned table
TABLE_NAME	VARCHAR2(30)		Name of the partitioned table
PARTITIONING_TYPE	VARCHAR2(9)		Type of the partitioning method: - RANGE - HASH - SYSTEM - LIST - REFERENCE
SUBPARTITIONING_TYPE	VARCHAR2(7)		Type of the composite partitioning method: - NONE - RANGE - HASH - SYSTEM - LIST
PARTITION_COUNT	NUMBER		Number of partitions in the table
DEF_SUBPARTITION_COUNT	NUMBER		For a composite-partitioned table, the default number of subpartitions, if specified
PARTITIONING_KEY_COUNT	NUMBER		Number of columns in the partitioning key

Column	Datatype	NULL	Description
SUBPARTITIONING_KEY_COUNT	NUMBER		For a composite-partitioned table, the number of columns in the subpartitioning key
STATUS	VARCHAR2(8)		If a previous DROP TABLE operation failed, indicates whether the table is unusable (UNUSABLE) or valid (VALID)
DEF_TABLESPACE_NAME	VARCHAR2(30)		Default tablespace to be used when adding a partition
DEF_PCT_FREE	NUMBER		Default value of PCTFREE to be used when adding a partition
DEF_PCT_USED	NUMBER		Default value of PCTUSED to be used when adding a partition
DEF_INI_TRANS	NUMBER		Default value of INITRANS to be used when adding a partition
DEF_MAX_TRANS	NUMBER		Default value of MAXTRANS to be used when adding a partition
DEF_INITIAL_EXTENT	VARCHAR2(40)		Default value of INITIAL (in Oracle blocks) to be used when adding a partition, or DEFAULT if no INITIAL value was specified
DEF_NEXT_EXTENT	VARCHAR2(40)		Default value of NEXT (in Oracle blocks) to be used when adding a partition, or DEFAULT if no NEXT value was specified
DEF_MIN_EXTENTS	VARCHAR2(40)		Default value of MINEXTENTS to be used when adding a partition, or DEFAULT if no MINEXTENTS value was specified
DEF_MAX_EXTENTS	VARCHAR2(40)		Default value of MAXEXTENTS to be used when adding a partition, or DEFAULT if no MAXEXTENTS value was specified
DEF_MAX_SIZE	VARCHAR2(40)		Default value of MAXSIZE to be used when adding a partition, or DEFAULT if no MAXSIZE value was specified
DEF_PCT_INCREASE	VARCHAR2(40)		Default value of PCTINCREASE to be used when adding a partition, or DEFAULT if no PCTINCREASE value was specified
DEF_FREELISTS	NUMBER		Default value of FREELISTS to be used when adding a partition
DEF_FREELIST_GROUPS	NUMBER		Default value of FREELIST GROUPS to be used when adding a partition
DEF_LOGGING	VARCHAR2(7)		Default LOGGING attribute to be used when adding a partition: ■ NONE ■ YES ■ NO ■ UNKNOWN
DEF_COMPRESSION	VARCHAR2(8)		Default compression to be used when adding a partition: ■ NONE ■ ENABLED ■ DISABLED ■ UNKNOWN
DEF_BUFFER_POOL	VARCHAR2(7)		Default buffer pool to be used when adding a partition: ■ DEFAULT ■ KEEP ■ RECYCLE ■ NULL

Column	Datatype	NULL	Description
REF_PTN_CONSTRAINT_NAME	VARCHAR2(30)		Name of the partitioning referential constraint for reference-partitioned tables
INTERVAL	VARCHAR2(1000)		String of the interval value

See Also:

- "DBA_PART_TABLES" on page 5-8
- "USER_PART_TABLES" on page 5-121

ALL_PARTIAL_DROP_TABS

ALL_PARTIAL_DROP_TABS describes tables accessible to the current user that have partially completed DROP COLUMN operations. Such operations might have been interrupted by the user or by a system crash.

Related Views

- DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations.
- USER_PARTIAL_DROP_TABS describes tables in the schema of the current user that have partially completed DROP COLUMN operations. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

ALL_PENDING_CONV_TABLES

ALL_PENDING_CONV_TABLES describes the pending conversion tables (tables which are not upgraded to the latest type version) accessible to the current user.

Related Views

- DBA_PENDING_CONV_TABLES describes all pending conversion tables in the database.
- USER_PENDING_CONV_TABLES describes the pending conversion tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table

See Also:

- "DBA_PENDING_CONV_TABLES" on page 5-8
- "USER_PENDING_CONV_TABLES" on page 5-121

ALL_PLSQL_OBJECT_SETTINGS

ALL_PLSQL_OBJECT_SETTINGS displays information about the compiler settings for the stored objects accessible to the current user.

Related Views

- DBA_PLSQL_OBJECT_SETTINGS displays information about the compiler settings for all stored objects in the database.
- USER_PLSQL_OBJECT_SETTINGS displays information about the compiler settings for the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY
PLSQL_OPTIMIZE_LEVEL	NUMBER		Optimization level that was used to compile the object
PLSQL_CODE_TYPE	VARCHAR2(4000)		Compilation mode for the object
PLSQL_DEBUG	VARCHAR2(4000)		Indicates whether the object was compiled with debug information or not
PLSQL_WARNINGS	VARCHAR2(4000)		Compiler warning settings that were used to compile the object
NLS_LENGTH_SEMANTICS	VARCHAR2(4000)		NLS length semantics that were used to compile the object
PLSQL_CCFLAGS	VARCHAR2(4000)		Conditional compilation flag settings that were used to compile the object
PLSCOPE_SETTINGS	VARCHAR2(4000)		Settings for using PL/Scope

See Also:

- "DBA_PLSQL_OBJECT_SETTINGS" on page 5-8
- "USER_PLSQL_OBJECT_SETTINGS" on page 5-121

ALL_POLICIES

ALL_POLICIES describes all Oracle Virtual Private Database (VPD) security policies for objects accessible to the current user. A security policy is a list of security requirements and rules that regulate access to those database objects.

Related Views

- DBA_POLICIES describes all Oracle Virtual Private Database (VPD) security policies in the database.

- USER_POLICIES describes all Oracle Virtual Private Database (VPD) security policies associated with objects owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
POLICY_GROUP	VARCHAR2(30)	NOT NULL	Name of the policy group
POLICY_NAME	VARCHAR2(30)	NOT NULL	Name of the policy
PF_OWNER	VARCHAR2(30)	NOT NULL	Owner of the policy function
PACKAGE	VARCHAR2(30)		Name of the package containing the policy function
FUNCTION	VARCHAR2(30)	NOT NULL	Name of the policy function
SEL	VARCHAR2(3)		Indicates whether the policy is applied to queries on the object (YES) or not (NO)
INS	VARCHAR2(3)		Indicates whether the policy is applied to INSERT statements on the object (YES) or not (NO)
UPD	VARCHAR2(3)		Indicates whether the policy is applied to UPDATE statements on the object (YES) or not (NO)
DEL	VARCHAR2(3)		Indicates whether the policy is applied to DELETE statements on the object (YES) or not (NO)
IDX	VARCHAR2(3)		Indicates whether the policy is enforced for index maintenance on the object (YES) or not (NO)
CHK_OPTION	VARCHAR2(3)		Indicates whether the check option is enforced for the policy (YES) or not (NO)
ENABLE	VARCHAR2(3)		Indicates whether the policy is enabled (YES) or disabled (NO)
STATIC_POLICY	VARCHAR2(3)		Indicates whether the policy is static (YES) or not (NO)
POLICY_TYPE	VARCHAR2(24)		Policy type: ■ STATIC ■ SHARED_STATIC ■ CONTEXT_SENSITIVE ■ SHARED_CONTEXT_SENSITIVE ■ DYNAMIC
LONG_PREDICATE	VARCHAR2(3)		Indicates whether the policy function can return a maximum of 32 KB of predicate (YES) or not (NO). If NO, the default maximum predicate size is 4000 bytes.

See Also:

- "DBA_POLICIES" on page 5-8
- "USER_POLICIES" on page 5-121
- *Oracle Database Concepts* for an overview of security policies and fine-grained access control
- *Oracle Database Security Guide* for information about implementing security policies
- The DBMS_RLS package in *Oracle Database PL/SQL Packages and Types Reference* for information on administering security policies

ALL_POLICY_CONTEXTS

ALL_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views accessible to the current user.

Related Views

- DBA_POLICY_CONTEXTS describes all driving contexts in the database.
- USER_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
NAMESPACE	VARCHAR2(30)	NOT NULL	Namespace of the driving context
ATTRIBUTE	VARCHAR2(30)	NOT NULL	Attribute of the driving context

See Also:

- "DBA_POLICY_CONTEXTS" on page 5-8
- "USER_POLICY_CONTEXTS" on page 5-122

ALL_POLICY_GROUPS

ALL_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views accessible to the current user.

Related Views

- DBA_POLICY_GROUPS describes all policy groups in the database.
- USER_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym, table, or view
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym, table, or view
POLICY_GROUP	VARCHAR2(30)	NOT NULL	Name of the policy group

See Also:

- "DBA_POLICY_GROUPS" on page 5-9
- "USER_POLICY_GROUPS" on page 5-122

ALL_PROCEDURES

ALL_PROCEDURES lists all functions and procedures, along with associated properties. For example, ALL_PROCEDURES indicates whether or not a function is pipelined, parallel enabled or an aggregate function. If a function is pipelined or an aggregate function, the associated implementation type (if any) is also identified.

Related Views

- DBA_PROCEDURES lists all functions and procedures, along with associated properties.
- USER_PROCEDURES lists all functions and procedures, along with associated properties. It does not contain the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the procedure
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object: top-level function, procedure, or package name
PROCEDURE_NAME	VARCHAR2(30)		Name of the procedure
OBJECT_ID	NUMBER		Object number of the object
SUBPROGRAM_ID	NUMBER		Unique subprogram identifier
OVERLOAD	VARCHAR2(40)		Overload unique identifier
OBJECT_TYPE	VARCHAR2(19)		The typename of the object
AGGREGATE	VARCHAR2(3)		Indicates whether the procedure is an aggregate function (YES) or not (NO)
PIPELINED	VARCHAR2(3)		Indicates whether the procedure is a pipelined table function (YES) or not (NO)
IMPLTYPEOWNER	VARCHAR2(30)		Name of the owner of the implementation type, if any
IMPLTYPENAME	VARCHAR2(30)		Name of the implementation type, if any
PARALLEL	VARCHAR2(3)		Indicates whether the procedure or function is parallel-enabled (YES) or not (NO)
INTERFACE	VARCHAR2(3)		YES, if the procedure/function is a table function implemented using the ODCI interface; otherwise NO
DETERMINISTIC	VARCHAR2(3)		YES, if the procedure/function is declared to be deterministic; otherwise NO
AUTHID	VARCHAR2(12)		Indicates whether the procedure/function is declared to execute as DEFINER or CURRENT_USER (invoker)

ALL_PROPAGATION

ALL_PROPAGATION displays information about the Streams propagations that have a source queue accessible to the current user.

Related View

DBA_PROPAGATION displays information about all Streams propagations in the database.

Column	Datatype	NULL	Description
PROPAGATION_NAME	VARCHAR2(30)	NOT NULL	Name of the Streams propagation
SOURCE_QUEUE_OWNER	VARCHAR2(30)		Owner of the source queue of the propagation
SOURCE_QUEUE_NAME	VARCHAR2(30)		Name of the source queue of the propagation
DESTINATION_QUEUE_OWNER	VARCHAR2(30)		Owner of the destination queue of the propagation
DESTINATION_QUEUE_NAME	VARCHAR2(30)		Name of the destination queue of the propagation
DESTINATION_DBLINK	VARCHAR2(128)		Database link to propagate events from the source queue to the destination queue
RULE_SET_OWNER	VARCHAR2(30)		Owner of the propagation positive rule set

Column	Datatype	NULL	Description
RULE_SET_NAME	VARCHAR2(30)		Name of the propagation positive rule set
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the propagation negative rule set
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the propagation negative rule set
QUEUE_TO_QUEUE	VARCHAR2(5)		Indicates whether the propagation is a queue-to-queue propagation (TRUE) or not (FALSE). A queue-to-queue propagation always has its own exclusive propagation job to propagate messages from the source queue to the destination queue.
STATUS	VARCHAR2(8)		Status of the propagation: ■ DISABLED ■ ENABLED ■ ABORTED
ERROR_MESSAGE	VARCHAR2(4000)		Error message last encountered by propagation
ERROR_DATE	DATE		Time that propagation last encountered an error
ORIGINAL_PROPAGATION_NAME	VARCHAR2(30)		Original propagation from which the propagation is cloned
ORIGINAL_SOURCE_QUEUE_OWNER	VARCHAR2(30)		Source queue owner of the original propagation
ORIGINAL_SOURCE_QUEUE_NAME	VARCHAR2(30)		Source queue name of the original propagation
ACKED_SCN	NUMBER		Acknowledged SCN of the subscribers of captured messages in the destination queue for the propagation
AUTO_MERGE_THRESHOLD	NUMBER		Merge threshold value for merging the propagation back to the original source queue. This is used as the value for the merge_threshold parameter in the DBMS_STREAMS_ADM.MERGE_STREAMS_JOB procedure.

See Also: "DBA_PROPAGATION" on page 5-9

ALL_PUBLISHED_COLUMNS

ALL_PUBLISHED_COLUMNS is a synonym for USER_PUBLISHED_COLUMNS.

See Also: "USER_PUBLISHED_COLUMNS" on page 5-122

ALL_QUEUE_SUBSCRIBERS

ALL_QUEUE_SUBSCRIBERS returns the list of subscribers that the current user has privilege to dequeue from.

Related Views

- DBA_QUEUE_SUBSCRIBERS returns the list of subscribers on all queues in the database.
- USER_QUEUE_SUBSCRIBERS returns the list of subscribers on queues that are under the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue

Column	Datatype	NULL	Description
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the queue table on which the queue is defined
CONSUMER_NAME	VARCHAR2(30)		Name of the subscriber
ADDRESS	VARCHAR2(1024)		Address of the subscriber
PROTOCOL	NUMBER		Protocol of the subscriber
TRANSFORMATION	VARCHAR2(61)		Transformation for the subscriber
DELIVERY_MODE	VARCHAR2(22)		Message delivery mode for the subscriber. There are three possible modes: PERSISTENT, BUFFERED, or PERSISTENT_OR_BUFFERED.
QUEUE_TO_QUEUE	VARCHAR2(5)		TRUE if the subscriber is a queue-to-queue subscriber; FALSE otherwise

ALL_QUEUE_TABLES

ALL_QUEUE_TABLES describes the queues in the queue tables accessible to the current user.

Related Views

- DBA_QUEUE_TABLES describes the queues in all queue tables in the database.
- USER_QUEUE_TABLES describes the queues in the queue tables created in the current user's schema. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema of the queue table
QUEUE_TABLE	VARCHAR2(30)		Name of the queue table
TYPE	VARCHAR2(7)		Type of user data: ■ RAW - Raw type ■ OBJECT - User-defined object type ■ VARIANT - Variant type (internal use only)
OBJECT_TYPE	VARCHAR2(61)		Object type of the payload when TYPE is OBJECT
SORT_ORDER	VARCHAR2(22)		User-specified sort order
RECIPIENTS	VARCHAR2(8)		SINGLE or MULTIPLE recipients
MESSAGE_GROUPING	VARCHAR2(13)		NONE or TRANSACTIONAL
COMPATIBLE	VARCHAR2(5)		Lowest release level which the queue table is compatible with (for example, 8.0.3)
PRIMARY_INSTANCE	NUMBER		Indicates the instance number of the instance which is the primary owner of the queue table. A value of 0 indicates that there is no primary owner.
SECONDARY_INSTANCE	NUMBER		Indicates the instance number of the instance which is the secondary owner of the queue table. This instance becomes the owner of the queue table if the primary owner is not alive. A value of 0 indicates that there is no secondary owner.
OWNER_INSTANCE	NUMBER		Instance number of the instance which currently owns the queue table
USER_COMMENT	VARCHAR2(50)		Comment supplied by the user
SECURE	VARCHAR2(3)		Indicates whether the queue table is secure (YES) or not (NO)

See Also:

- "DBA_QUEUE_TABLES" on page 5-12
- "USER_QUEUE_TABLES" on page 5-123
- *Oracle Streams Advanced Queuing User's Guide* for more information about these views and Advanced Queuing

ALL_QUEUES

ALL_QUEUES describes all queues on which the current user has enqueue or dequeue privileges. If the user has any Advanced Queuing system privileges, like MANAGE ANY QUEUE, ENQUEUE ANY QUEUE or DEQUEUE ANY QUEUE, then this view describes all queues in the database.

Related Views

- DBA_QUEUES describes all queues in the database.
- USER_QUEUES describes the operational characteristics of every queue owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_TABLE	VARCHAR2(30)	NOT NULL	Name of the table the queue data resides in
QID	NUMBER	NOT NULL	Object number of the queue
QUEUE_TYPE	VARCHAR2(20)		Type of the queue: ■ EXCEPTION_QUEUE ■ NON_PERSISTENT_QUEUE ■ NORMAL_QUEUE
MAX_RETRIES	NUMBER		Maximum number of retries allowed when dequeuing from the queue
RETRY_DELAY	NUMBER		Time interval between retries
ENQUEUE_ENABLED	VARCHAR2(7)		Indicates whether the queue is enabled for enqueue (YES) or not (NO)
DEQUEUE_ENABLED	VARCHAR2(7)		Indicates whether the queue is enabled for dequeue (YES) or not (NO)
RETENTION	VARCHAR2(40)		Time interval (in seconds) processed messages are retained in the queue, or FOREVER
USER_COMMENT	VARCHAR2(50)		User specified comment
NETWORK_NAME	VARCHAR2(64)		Network name

See Also: *Oracle Streams Advanced Queuing User's Guide* for more information about these views and Advanced Queuing

ALL_REFRESH

ALL_REFRESH describes all the refresh groups accessible to the current user.

Related Views

- DBA_REFRESH describes all refresh groups in the database.

- USER_REFRESH describes all refresh groups owned by the current user.

Column	Datatype	NULL	Description
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		(Y |N) If Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y |N) If Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		(Y |N) Y means the job is broken and will never be run
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

ALL_REFRESH_CHILDREN

ALL_REFRESH_CHILDREN lists all the objects in refresh groups that are accessible to the current user.

Related Views

- DBA_REFRESH_CHILDREN describes the objects in all refresh groups in the database.
- USER_REFRESH_CHILDREN describes the objects in all refresh groups owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE	VARCHAR2(30)		Type of the object in the refresh group
ROWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the refresh group
RNAME	VARCHAR2(30)	NOT NULL	Name of the refresh group
REFGROUP	NUMBER		Internal identifier of refresh group
IMPLICIT_DESTROY	VARCHAR2(1)		(Y| N) If Y, then destroy the refresh group when its last item is subtracted
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y |N) If Y then push changes from snapshot to master before refresh
REFRESH_AFTER _ERRORS	VARCHAR2(1)		If Y, proceed with refresh despite error when pushing deferred RPCs
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing

Column	Datatype	NULL	Description
JOB	NUMBER		Identifier of job used to refresh the group automatically
NEXT_DATE	DATE		Date that this job will next be refreshed automatically, if not broken
INTERVAL	VARCHAR2(200)		A date function used to compute the next NEXT_DATE
BROKEN	VARCHAR2(1)		(Y |N) Y means the job is broken and will never be run
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

ALL_REFRESH_DEPENDENCIES

ALL_REFRESH_DEPENDENCIES lists the names of the dependent detail or container tables of all the materialized views in the current schema.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name, unique within this schema
PARENT_OBJECT_TYPE	CHAR(17)		MATERIALIZED VIEW
OLDEST_REFRESH_SCN	NUMBER		The minimum SCN of any summary or materialized view that has TABLE_NAME as a detail table
OLDEST_REFRESH_DATE	DATE		SYSDATE when last refreshed

ALL_REFS

ALL_REFS describes the REF columns and REF attributes in object type columns accessible to the current user.

Related Views

- DBA_REFS describes all REF columns and REF attributes in the database.
- USER_REFS describes the REF columns and REF attributes in object type columns owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COLUMN_NAME	VARCHAR2(4000)		Name of the REF column or attribute. If it is not a top-level attribute, the value of COLUMN_NAME should be a path name starting with the column name
WITH_ROWID	VARCHAR2(3)		Indicates whether the REF value is stored with ROWID (YES) or not (NO)
IS_SCOPED	VARCHAR2(3)		Indicates whether the REF column is scoped (YES) or not (NO)
SCOPE_TABLE_OWNER	VARCHAR2(30)		Name of the owner of the scope table, if it exists and is accessible by the user
SCOPE_TABLE_NAME	VARCHAR2(30)		Name of the scope table, if it exists and is accessible by the user
OBJECT_ID_TYPE	VARCHAR2(16)		If user-defined OID, then USER-DEFINED, else if system generated OID, then SYSTEM GENERATED

ALL_REGISTERED_MVIEWS

ALL_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) accessible to the current user.

Related Views

- DBA_REGISTERED_MVIEWS describes all registered materialized views in the database.
- USER_REGISTERED_MVIEWS describes all registered materialized views owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the materialized view
NAME	VARCHAR2(30)	NOT NULL	Name of the materialized view
MVIEW_SITE	VARCHAR2(128)	NOT NULL	Global name of the materialized view site
CAN_USE_LOG	VARCHAR2(3)		YES if the materialized view can use a materialized view log, NO if the materialized view is too complex to use a log.
UPDATABLE	VARCHAR2(3)		(YES/NO) Indicates whether the materialized view is updatable. If set to NO, then the materialized view is read only.
REFRESH_METHOD	VARCHAR2(11)		Whether the materialized view uses primary key, rowids, or object identifiers for fast refresh
MVIEW_ID	NUMBER(38)		Identifier for the materialized view used by the masters for fast refresh
VERSION	VARCHAR2(26)		Oracle version of the materialized view **Note:** Oracle Database materialized views show ORACLE 8 MATERIALIZED VIEW.
QUERY_TXT	LONG		Query that defines the materialized view

See Also:

- "DBA_REGISTERED_MVIEWS" on page 5-17
- "USER_REGISTERED_MVIEWS" on page 5-123

ALL_REGISTRY_BANNERS

ALL_REGISTRY_BANNERS displays the valid components loaded into the database.

Column	Datatype	NULL	Description
BANNER	VARCHAR2(80)		Component display banner

ALL_REWRITE_EQUIVALENCES

ALL_REWRITE_EQUIVALENCES describes the rewrite equivalences accessible to the current user.

Related Views

- DBA_REWRITE_EQUIVALENCES describes all rewrite equivalences in the database.

- USER_REWRITE_EQUIVALENCES describes the rewrite equivalences owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the rewrite equivalence
NAME	VARCHAR2(30)	NOT NULL	Name of the rewrite equivalence
SOURCE_STMT	CLOB		Source statement of the rewrite equivalence
DESTINATION_STMT	CLOB		Destination of the rewrite equivalence
REWRITE_MODE	VARCHAR2(10)		Rewrite mode of the rewrite equivalence: ■ DISABLED ■ TEXT_MATCH ■ GENERAL ■ RECURSIVE

See Also:

- "DBA_REWRITE_EQUIVALENCES" on page 5-21
- "USER_REWRITE_EQUIVALENCES" on page 5-124

ALL_RULE_SET_RULES

ALL_RULE_SET_RULES describes the rules in the rule sets accessible to the current user.

Related Views

- DBA_RULE_SET_RULES describes the rules in all rule sets in the database.
- USER_RULE_SET_RULES describes the rules in the rule sets owned by the current user. This view does not display the RULE_SET_OWNER column.

Column	Datatype	NULL	Description
RULE_SET_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule set
RULE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the rule set
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_SET_RULE_ENABLED	VARCHAR2(8)		Indicates whether the rule is enabled in the rule set (ENABLED) or not (DISABLED)
RULE_SET_RULE_EVAL_CTX_OWNER	VARCHAR2(30)		Owner of the evaluation context specified when the rule was added to the rule set, if any
RULE_SET_RULE_EVAL_CTX_NAME	VARCHAR2(30)		Name of the evaluation context specified when the rule was added to the rule set, if any
RULE_SET_RULE_COMMENT	VARCHAR2(4000)		Comment specified when the rule was added to the rule set, if any

See Also:

- "DBA_RULE_SET_RULES" on page 5-27
- "USER_RULE_SET_RULES" on page 5-125

ALL_RULE_SETS

ALL_RULE_SETS describes the rule sets accessible to the current user.

Related Views

- DBA_RULE_SETS describes all rule sets in the database.
- USER_RULE_SETS describes the rule sets owned by the current user. This view does not display the RULE_SET_OWNER column.

Column	Datatype	NULL	Description
RULE_SET_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule set
RULE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the rule set
RULE_SET_EVAL_CONTEXT_OWNER	VARCHAR2(30)		Owner of the evaluation context associated with the rule set, if any
RULE_SET_EVAL_CONTEXT_NAME	VARCHAR2(30)		Name of the evaluation context associated with the rule set, if any
RULE_SET_COMMENT	VARCHAR2(4000)		Comment specified with the rule set, if any

See Also:

- "DBA_RULE_SETS" on page 5-27
- "USER_RULE_SETS" on page 5-125

ALL_RULES

ALL_RULES describes the rules accessible to the current user.

Related Views

- DBA_RULES describes all rules in the database.
- USER_RULES describes the rules owned by the current user. This view does not display the RULE_OWNER column.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_CONDITION	CLOB		Expressions and operators that constitute the rule condition
RULE_EVALUATION_CONTEXT_OWNER	VARCHAR2(30)		Owner of the evaluation context associated with the rule, if any
RULE_EVALUATION_CONTEXT_NAME	VARCHAR2(30)		Name of the evaluation context associated with the rule, if any
RULE_ACTION_CONTEXT	RE$NV_LIST		Action context associated with the rule, if any
RULE_COMMENT	VARCHAR2(4000)		Comment specified with the rule, if any

See Also:

- "DBA_RULES" on page 5-28
- "USER_RULES" on page 5-125

ALL_SCHEDULER_CHAIN_RULES

ALL_SCHEDULER_CHAIN_RULES displays information about the rules for the chains accessible to the current user (that is, those chains that the user has ALTER or EXECUTE privileges for).

Related Views

- DBA_SCHEDULER_CHAIN_RULES displays information about the rules for all chains in the database.
- USER_SCHEDULER_CHAIN_RULES displays information about the rules for the chains owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler chain that the rule is in
CHAIN_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler chain that the rule is in
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
CONDITION	VARCHAR2(4000)		Boolean condition triggering the rule
ACTION	VARCHAR2(4000)		Action to be performed when the rule is triggered
COMMENTS	VARCHAR2(4000)		User-specified comments about the rule

See Also:

- "DBA_SCHEDULER_CHAIN_RULES" on page 5-28
- "USER_SCHEDULER_CHAIN_RULES" on page 5-125

ALL_SCHEDULER_CHAIN_STEPS

ALL_SCHEDULER_CHAIN_STEPS displays information about the defined steps of the chains accessible to the current user (that is, those chains that the user has ALTER or EXECUTE privileges for).

Related Views

- DBA_SCHEDULER_CHAIN_STEPS displays information about the defined steps of all chains in the database.
- USER_SCHEDULER_CHAIN_STEPS displays information about the defined steps of the chains owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler chain the step is in
CHAIN_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler chain the step is in
STEP_NAME	VARCHAR2(30)	NOT NULL	Name of the chain step
PROGRAM_OWNER	VARCHAR2(98)		Owner of the program that runs during the step
PROGRAM_NAME	VARCHAR2(98)		Name of the program that runs during the step
EVENT_SCHEDULE_OWNER	VARCHAR2(98)		Owner of the event schedule that this step waits for
EVENT_SCHEDULE_NAME	VARCHAR2(98)		Name of the event schedule that this step waits for
EVENT_QUEUE_OWNER	VARCHAR2(30)		Owner of the source queue into which the event will be raised

Column	Datatype	NULL	Description
EVENT_QUEUE_NAME	VARCHAR2(30)		Name of the source queue into which the event will be raised
EVENT_QUEUE_AGENT	VARCHAR2(30)		Name of the AQ agent used by the user on the event source queue (for a secure queue)
EVENT_CONDITION	VARCHAR2(4000)		Boolean expression used as the subscription rule for an event on the source queue
CREDENTIAL_OWNER	VARCHAR2(30)		Owner of the credential to be used for an external step job
CREDENTIAL_NAME	VARCHAR2(30)		Name of the credential to be used for an external step job
DESTINATION	VARCHAR2(128)		Destination host on which a remote step job will run
SKIP	VARCHAR2(5)		Indicates whether the step should be skipped (TRUE) or not (FALSE)
PAUSE	VARCHAR2(5)		Indicates whether the step should be paused after running (TRUE) or not (FALSE)
RESTART_ON_RECOVERY	VARCHAR2(5)		Indicates whether the step should be restarted on database recovery (TRUE) or not (FALSE)
STEP_TYPE	VARCHAR2(14)		Type of the step: ■ EVENT_SCHEDULE ■ INLINE_EVENT ■ SUBCHAIN ■ PROGRAM
TIMEOUT	INTERVAL DAY(3) TO SECOND(0)		Timeout for waiting on an event schedule

See Also:

- "DBA_SCHEDULER_CHAIN_STEPS" on page 5-28
- "USER_SCHEDULER_CHAIN_STEPS" on page 5-125

ALL_SCHEDULER_CHAINS

ALL_SCHEDULER_CHAINS displays information about the chains accessible to the current user (that is, those chains that the user has ALTER or EXECUTE privileges for).

Related Views

- DBA_SCHEDULER_CHAINS displays information about all chains in the database.
- USER_SCHEDULER_CHAINS displays information about the chains owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler chain
CHAIN_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler chain
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set describing the dependencies
RULE_SET_NAME	VARCHAR2(30)		Name of the rule set describing the dependencies
NUMBER_OF_RULES	NUMBER		Number of rules in the chain
NUMBER_OF_STEPS	NUMBER		Number of defined steps in the chain
ENABLED	VARCHAR2(5)		Indicates whether the chain is enabled (TRUE) or disabled (FALSE)

Column	Datatype	NULL	Description
EVALUATION_INTERVAL	INTERVAL DAY(3) TO SECOND(0)		Periodic interval at which to reevaluate rules for the chain
USER_RULE_SET	VARCHAR2(5)		Indicates whether the chain uses a user-specified rule set (TRUE) or not (FALSE)
COMMENTS	VARCHAR2(240)		Comments on the chain

See Also:

- "DBA_SCHEDULER_CHAINS" on page 5-28
- "USER_SCHEDULER_CHAINS" on page 5-125

ALL_SCHEDULER_CREDENTIALS

ALL_SCHEDULER_CREDENTIALS displays information about the credentials accessible to the current user (that is, those credentials that the user has ALTER or EXECUTE privileges for).

Related Views

- DBA_SCHEDULER_CREDENTIALS displays information about all credentials in the database.
- USER_SCHEDULER_CREDENTIALS displays information about the credentials owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler credential
CREDENTIAL_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler credential
USERNAME	VARCHAR2(64)	NOT NULL	Name of the user that will be used to log in to the remote database or operating system
DATABASE_ROLE	VARCHAR2(7)		For a database target, the database role to use when logging in: ■ SYSDBA ■ SYSOPER
WINDOWS_DOMAIN	VARCHAR2(30)		For a Windows target, the Windows domain to use when logging in
COMMENTS	VARCHAR2(240)		Comments on the credential

See Also:

- "DBA_SCHEDULER_CREDENTIALS" on page 5-28
- "USER_SCHEDULER_CREDENTIALS" on page 5-125

ALL_SCHEDULER_GLOBAL_ATTRIBUTE

ALL_SCHEDULER_GLOBAL_ATTRIBUTE displays the values of all scheduler attributes (for example, DEFAULT_TIMEZONE and CURRENT_OPEN_WINDOW).

Related View

DBA_SCHEDULER_GLOBAL_ATTRIBUTE displays the values of all scheduler attributes in the database.

Column	Datatype	NULL	Description
ATTRIBUTE_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler attribute
VALUE	VARCHAR2(128)		Value of the Scheduler attribute

See Also: "DBA_SCHEDULER_GLOBAL_ATTRIBUTE" on page 5-28

ALL_SCHEDULER_JOB_ARGS

ALL_SCHEDULER_JOB_ARGS displays information about the arguments of the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOB_ARGS displays information about the arguments of all Scheduler jobs in the database.
- USER_SCHEDULER_JOB_ARGS displays information about the arguments of the Scheduler jobs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the job to which the argument belongs
JOB_NAME	VARCHAR2(30)		Name of the job to which the argument belongs
ARGUMENT_NAME	VARCHAR2(30)		Optional name of the argument
ARGUMENT_POSITION	NUMBER		Position of the argument in the argument list
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the argument
VALUE	VARCHAR2(4000)		Value of the argument (in string format) if the argument is a string
ANYDATA_VALUE	ANYDATA		Value of the argument (in AnyData format)
OUT_ARGUMENT	VARCHAR2(5)		Reserved for future use

See Also:

- "DBA_SCHEDULER_JOB_ARGS" on page 5-28
- "USER_SCHEDULER_JOB_ARGS" on page 5-126

ALL_SCHEDULER_JOB_CLASSES

ALL_SCHEDULER_JOB_CLASSES displays information about the Scheduler job classes accessible to the current user.

Related View

DBA_SCHEDULER_JOB_CLASSES displays information about all Scheduler job classes in the database.

Column	Datatype	NULL	Description
JOB_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler job class
RESOURCE_CONSUMER_GROUP	VARCHAR2(30)		Resource consumer group associated with the class
SERVICE	VARCHAR2(64)		Name of the service the class is associated with

Column	Datatype	NULL	Description
LOGGING_LEVEL	VARCHAR2(11)		Amount of logging that will be done pertaining to the class: ▪ OFF ▪ RUNS ▪ FAILED RUNS ▪ FULL
LOG_HISTORY	NUMBER		History (in days) to maintain in the job log for the class
COMMENTS	VARCHAR2(240)		Comments on the class

See Also: "DBA_SCHEDULER_JOB_CLASSES" on page 5-29

ALL_SCHEDULER_JOB_LOG

ALL_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOB_LOG displays log information for all Scheduler jobs in the database.
- USER_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs owned by the current user.

Column	Datatype	NULL	Description
LOG_ID	NUMBER	NOT NULL	Unique identifier that identifies a row
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
OWNER	VARCHAR2(30)		Owner of the Scheduler job
JOB_NAME	VARCHAR2(65)		Name of the Scheduler job
JOB_SUBNAME	VARCHAR2(65)		Subname of the Scheduler job (for a chain step job)
JOB_CLASS	VARCHAR2(30)		Class that the job belonged to at the time of entry
OPERATION	VARCHAR2(30)		Operation corresponding to the log entry

Column	Datatype	NULL	Description
STATUS	VARCHAR2(30)		Status of the operation, if applicable. Possible values for this column are dependent on the value in the OPERATION column. In most cases, STATUS will be NULL. Only for job run operations will it have a value. STATUS will be NULL when OPERATION is one of the following: ■ CREATE - Job was created ■ UPDATE - One or more job attributes have been modified ■ ENABLE - Job has been enabled ■ DISABLE - Job has been disabled ■ COMPLETED - For repeating jobs only, job has reached its end date or maximum number of runs ■ BROKEN - Job has reached its maximum number of failures STATUS can be SUCCEEDED (job run completed successfully), FAILED (job run failed), or STOPPED (job run was stopped) when OPERATION is one of the following: ■ RUN - Regular job run ■ RETRY_RUN - Job is being retried because the previous run resulted in an error and RESTARTABLE is set to TRUE ■ RECOVERY_RUN - Job is being rerun because the database went down, or the job slave crashed and RESTARTABLE is set to TRUE
USER_NAME	VARCHAR2(30)		Name of the user who performed the operation, if applicable
CLIENT_ID	VARCHAR2(64)		Client identifier of the user who performed the operation, if applicable
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user who performed the operation, if applicable
DESTINATION	VARCHAR2(128)		Destination for a remote job operation
ADDITIONAL_INFO	CLOB		Additional information on the entry, if applicable

See Also:

- "DBA_SCHEDULER_JOB_LOG" on page 5-29
- "USER_SCHEDULER_JOB_LOG" on page 5-126

ALL_SCHEDULER_JOB_RUN_DETAILS

ALL_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOB_RUN_DETAILS displays log run details for all Scheduler jobs in the database.
- USER_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs owned by the current user.

Column	Datatype	NULL	Description
LOG_ID	NUMBER		Unique identifier of the log entry (foreign key of the *_SCHEDULER_JOB_LOG views)

Column	Datatype	NULL	Description
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
OWNER	VARCHAR2(30)		Owner of the Scheduler job
JOB_NAME	VARCHAR2(65)		Name of the Scheduler job
JOB_SUBNAME	VARCHAR2(65)		Subname of the Scheduler job (for a chain step job)
STATUS	VARCHAR2(30)		Status of the job run
ERROR#	NUMBER		Error number in the case of an error
REQ_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Requested start date of the job run
ACTUAL_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Actual date on which the job was run
RUN_DURATION	INTERVAL DAY(3) TO SECOND(0)		Duration of the job run
INSTANCE_ID	NUMBER		Identifier of the instance on which the job was run
SESSION_ID	VARCHAR2(30)		Session identifier of the job run
SLAVE_PID	VARCHAR2(30)		Process identifier of the slave on which the job was run
CPU_USED	INTERVAL DAY(3) TO SECOND(2)		Amount of CPU used for the job run
DESTINATION	VARCHAR2(128)		Destination for a remote job operation
ADDITIONAL_INFO	VARCHAR2(4000)		Additional information on the job run, if applicable

See Also:

- "DBA_SCHEDULER_JOB_RUN_DETAILS" on page 5-30
- "USER_SCHEDULER_JOB_RUN_DETAILS" on page 5-126

ALL_SCHEDULER_JOBS

ALL_SCHEDULER_JOBS displays information about the Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_JOBS displays information about all Scheduler jobs in the database.
- USER_SCHEDULER_JOBS displays information about the Scheduler jobs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)		Name of the Scheduler job
JOB_SUBNAME	VARCHAR2(30)		Subname of the Scheduler job (for a job running a chain step)
JOB_STYLE	VARCHAR2(11)		Job style: ■ REGULAR ■ LIGHTWEIGHT
JOB_CREATOR	VARCHAR2(30)		Original creator of the job
CLIENT_ID	VARCHAR2(64)		Client identifier of the user creating the job
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user creating the job

Column	Datatype	NULL	Description
PROGRAM_OWNER	VARCHAR2(4000)		Owner of the program associated with the job
PROGRAM_NAME	VARCHAR2(4000)		Name of the program associated with the job
JOB_TYPE	VARCHAR2(16)		Inline job action type: ■ PLSQL_BLOCK ■ STORED_PROCEDURE ■ EXECUTABLE ■ CHAIN
JOB_ACTION	VARCHAR2(4000)		Inline job action
NUMBER_OF_ARGUMENTS	NUMBER		Inline number of job arguments
SCHEDULE_OWNER	VARCHAR2(4000)		Owner of the schedule that the job uses (can be a window or a window group)
SCHEDULE_NAME	VARCHAR2(4000)		Name of the schedule that the job uses (can be a window or a window group)
SCHEDULE_TYPE	VARCHAR2(12)		Type of the schedule that the job uses: ■ IMMEDIATE - Start date and repeat interval are NULL ■ ONCE - Repeat interval is NULL ■ PLSQL - PL/SQL expression used as schedule ■ CALENDAR - Oracle calendaring expression used as schedule ■ EVENT - Event schedule ■ NAMED - Named schedule ■ WINDOW - Window used as schedule ■ WINDOW_GROUP - Window group used as schedule
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Original scheduled start date of the job (for an inline schedule)
REPEAT_INTERVAL	VARCHAR2(4000)		Inline schedule PL/SQL expression or calendar string
EVENT_QUEUE_OWNER	VARCHAR2(30)		Owner of the source queue into which the event will be raised
EVENT_QUEUE_NAME	VARCHAR2(30)		Name of the source queue into which the event will be raised
EVENT_QUEUE_AGENT	VARCHAR2(256)		Name of the AQ agent used by the user on the event source queue (if it is a secure queue)
EVENT_CONDITION	VARCHAR2(4000)		Boolean expression used as the subscription rule for the event on the source queue
EVENT_RULE	VARCHAR2(65)		Name of the rule used by the coordinator to trigger the event-based job
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Date after which the job will no longer run (for an inline schedule)
JOB_CLASS	VARCHAR2(30)		Name of the job class associated with the job
ENABLED	VARCHAR2(5)		Indicates whether the job is enabled (TRUE) or disabled (FALSE)
AUTO_DROP	VARCHAR2(5)		Indicates whether the job will be dropped when it has completed (TRUE) or not (FALSE)
RESTARTABLE	VARCHAR2(5)		Indicates whether the job can be restarted (TRUE) or not (FALSE)

Column	Datatype	NULL	Description
STATE	VARCHAR2(15)		Current state of the job: - DISABLED - RETRY SCHEDULED - SCHEDULED - RUNNING - COMPLETED - BROKEN - FAILED - REMOTE - SUCCEEDED - CHAIN_STALLED
JOB_PRIORITY	NUMBER		Priority of the job relative to other jobs in the same class
RUN_COUNT	NUMBER		Number of times the job has run
MAX_RUNS	NUMBER		Maximum number of times the job is scheduled to run
FAILURE_COUNT	NUMBER		Number of times the job has failed to run
MAX_FAILURES	NUMBER		Number of times the job will be allowed to fail before being marked broken
RETRY_COUNT	NUMBER		Number of times the job has retried, if it is retrying
LAST_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Last date on which the job started running
LAST_RUN_DURATION	INTERVAL DAY(9) TO SECOND(6)		Amount of time the job took to complete during the last run
NEXT_RUN_DATE	TIMESTAMP(6) WITH TIME ZONE		Next date on which the job is scheduled to run
SCHEDULE_LIMIT	INTERVAL DAY(3) TO SECOND(0)		Time after which a job which has not run yet will be rescheduled
MAX_RUN_DURATION	INTERVAL DAY(3) TO SECOND(0)		Maximum amount of time for which the job will be allowed to run
LOGGING_LEVEL	VARCHAR2(11)		Amount of logging that will be done pertaining to the job: - OFF - RUNS - FAILED RUNS - FULL
STOP_ON_WINDOW_CLOSE	VARCHAR2(5)		Indicates whether the job will stop if a window associated with the job closes (TRUE) or not (FALSE)
INSTANCE_STICKINESS	VARCHAR2(5)		Indicates whether the job is sticky (TRUE) or not (FALSE)
RAISE_EVENTS	VARCHAR2(4000)		List of job events to raise for the job: - JOB_STARTED - JOB_SUCCEEDED - JOB_FAILED - JOB_BROKEN - JOB_COMPLETED - JOB_STOPPED - JOB_SCH_LIM_REACHED - JOB_DISABLED - JOB_CHAIN_STALLED - JOB_OVER_MAX_DUR
SYSTEM	VARCHAR2(5)		Indicates whether the job is a system job (TRUE) or not (FALSE)

Column	Datatype	NULL	Description
JOB_WEIGHT	NUMBER		Weight of the job
NLS_ENV	VARCHAR2(4000)		NLS environment of the job
SOURCE	VARCHAR2(128)		Source global database identifier
DESTINATION	VARCHAR2(128)		Destination global database identifier
CREDENTIAL_OWNER	VARCHAR2(30)		Owner of the credential to be used for an external job
CREDENTIAL_NAME	VARCHAR2(30)		Name of the credential to be used for an external job
INSTANCE_ID	NUMBER		Instance on which the user requests the job to run
DEFERRED_DROP	VARCHAR2(5)		Indicates whether the job will be dropped when completed due to user request (TRUE) or not (FALSE)
COMMENTS	VARCHAR2(240)		Comments on the job
FLAGS	NUMBER		This column is for internal use

See Also:

- "DBA_SCHEDULER_JOBS" on page 5-30
- "USER_SCHEDULER_JOBS" on page 5-126

ALL_SCHEDULER_PROGRAM_ARGS

ALL_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs accessible to the current user.

Related Views

- DBA_SCHEDULER_PROGRAM_ARGS displays information about the arguments of all Scheduler programs in the database.
- USER_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the program to which the argument belongs
PROGRAM_NAME	VARCHAR2(30)	NOT NULL	Name of the program to which the argument belongs
ARGUMENT_NAME	VARCHAR2(30)		Optional name of the argument
ARGUMENT_POSITION	NUMBER	NOT NULL	Position of the argument in the argument list
ARGUMENT_TYPE	VARCHAR2(61)		Datatype of the argument
METADATA_ATTRIBUTE	VARCHAR2(19)		Metadata attribute: ■ JOB_NAME ■ JOB_OWNER ■ JOB_START ■ WINDOW_START ■ WINDOW_END ■ JOB_SUBNAME ■ EVENT_MESSAGE ■ JOB_SCHEDULED_START
DEFAULT_VALUE	VARCHAR2(4000)		Default value taken by the argument (in string format) if the argument is a string

Column	Datatype	NULL	Description
DEFAULT_ANYDATA_VALUE	ANYDATA		Default value taken by the argument (in AnyData format)
OUT_ARGUMENT	VARCHAR2(5)		Reserved for future use

See Also:

- "DBA_SCHEDULER_PROGRAM_ARGS" on page 5-30
- "USER_SCHEDULER_PROGRAM_ARGS" on page 5-126

ALL_SCHEDULER_PROGRAMS

ALL_SCHEDULER_PROGRAMS displays information about the Scheduler programs accessible to the current user.

Related Views

- DBA_SCHEDULER_PROGRAMS displays information about all Scheduler programs in the database.
- USER_SCHEDULER_PROGRAMS displays information about the Scheduler programs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler program
PROGRAM_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler program
PROGRAM_TYPE	VARCHAR2(16)		Type of the program action: ■ PLSQL_BLOCK ■ STORED_PROCEDURE ■ EXECUTABLE
PROGRAM_ACTION	VARCHAR2(4000)		String specifying the program action
NUMBER_OF_ARGUMENTS	NUMBER		Number of arguments accepted by the program
ENABLED	VARCHAR2(5)		Indicates whether the program is enabled (TRUE) or disabled (FALSE)
DETACHED	VARCHAR2(5)		This column is for internal use
SCHEDULE_LIMIT	INTERVAL DAY(3) TO SECOND(0)		Maximum delay in running the program after the scheduled start
PRIORITY	NUMBER		Priority of the program
WEIGHT	NUMBER		Weight of the program
MAX_RUNS	NUMBER		Maximum number of runs of any job based on this program
MAX_FAILURES	NUMBER		Maximum number of failures of any job based on this program
MAX_RUN_DURATION	INTERVAL DAY(3) TO SECOND(0)		Maximum amount of time this program can run
NLS_ENV	VARCHAR2(4000)		NLS environment in which the program was created
COMMENTS	VARCHAR2(240)		Comments on the program

See Also:

- "DBA_SCHEDULER_PROGRAMS" on page 5-30
- "USER_SCHEDULER_PROGRAMS" on page 5-126

ALL_SCHEDULER_REMOTE_DATABASES

ALL_SCHEDULER_REMOTE_DATABASES displays information about the remote databases accessible to the current user that have been registered as sources and destinations for remote database jobs.

Related View

DBA_SCHEDULER_REMOTE_DATABASES displays information about all remote databases that have been registered as sources and destinations for remote database jobs.

Column	Datatype	NULL	Description
DATABASE_NAME	VARCHAR2(512)	NOT NULL	Global name of the remote database
REGISTERED_AS	VARCHAR2(11)		Indicates whether the database is registered as a source (SOURCE) or as a destination (DESTINATION)
DATABASE_LINK	VARCHAR2(512)	NOT NULL	Name of a valid database link to the remote database

See Also: "DBA_SCHEDULER_REMOTE_DATABASES" on page 5-31

ALL_SCHEDULER_REMOTE_JOBSTATE

ALL_SCHEDULER_REMOTE_JOBSTATE displays information about the state of the jobs accessible to the current user at remote databases.

Related Views

- DBA_SCHEDULER_REMOTE_JOBSTATE displays information about the state of all jobs at remote databases.
- USER_SCHEDULER_REMOTE_JOBSTATE displays information about the state of the jobs owned by the current user at remote databases. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler job
DESTINATION	VARCHAR2(512)	NOT NULL	Name of the job destination

Column	Datatype	NULL	Description
STATE	VARCHAR2(15)		State of the job at the destination: ■ DISABLED ■ RETRY SCHEDULED ■ SCHEDULED ■ RUNNING ■ COMPLETED ■ BROKEN ■ FAILED ■ SUCCEEDED ■ STOPPED
NEXT_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Next start date of the job at the destination
RUN_COUNT	NUMBER		Run count of the job at the destination
FAILURE_COUNT	NUMBER		Failure count of the job at the destination
RETRY_COUNT	NUMBER		Retry count of the job at the destination
LAST_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Last start date of the job at the destination
LAST_END_DATE	TIMESTAMP(6) WITH TIME ZONE		Last end date of the job at the destination

See Also:

- "DBA_SCHEDULER_REMOTE_JOBSTATE" on page 5-31
- "USER_SCHEDULER_REMOTE_JOBSTATE" on page 5-126

ALL_SCHEDULER_RUNNING_CHAINS

ALL_SCHEDULER_RUNNING_CHAINS displays information about the chain steps of the running chains accessible to the current user (that is, those chains that the user has ALTER privileges for). In the case of nested chains, this view also enables you to traverse the hierarchy of the chain with a SQL statement that contains a CONNECT BY clause linking up the JOB_SUBNAME and STEP_JOB_SUBNAME columns.

Related Views

- DBA_SCHEDULER_RUNNING_CHAINS displays information about the chain steps of all running chains in the database.
- USER_SCHEDULER_RUNNING_CHAINS displays information about the chain steps of the running chains owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the job which is running the chain
JOB_NAME	VARCHAR2(30)	NOT NULL	Name of the job which is running the chain
JOB_SUBNAME	VARCHAR2(30)		Subname of the job which is running the chain (for a nested chain), else NULL
CHAIN_OWNER	VARCHAR2(30)	NOT NULL	Owner of the chain being run
CHAIN_NAME	VARCHAR2(30)	NOT NULL	Name of the chain being run
STEP_NAME	VARCHAR2(30)	NOT NULL	Name of the step of the running chain

Column	Datatype	NULL	Description
STATE	VARCHAR2(11)		State of the running chain step: - NOT_STARTED - RUNNING - SUCCEEDED - STOPPED - FAILED - SCHEDULED - PAUSED - STALLED
ERROR_CODE	NUMBER		Error code with which the step completed (if it has completed)
COMPLETED	VARCHAR2(5)		Indicates whether the running chain step has completed (TRUE) or not (FALSE)
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Date when the running chain step started (if it has started)
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Date when the running chain step stopped (if it has stopped)
DURATION	INTERVAL DAY(9) TO SECOND(6)		Amount of time it took the chain step to complete (if it has completed)
SKIP	VARCHAR2(5)		Indicates whether the chain step should be skipped (TRUE) or not (FALSE)
PAUSE	VARCHAR2(5)		Indicates whether the chain step should be paused after running (TRUE) or not (FALSE)
RESTART_ON_RECOVERY	VARCHAR2(5)		Indicates whether the chain step will be restarted on database recovery (TRUE) or not (FALSE)
STEP_JOB_SUBNAME	VARCHAR2(30)		Subname of the job running the step
STEP_JOB_LOG_ID	NUMBER		Log ID of the job running the step

See Also:

- "DBA_SCHEDULER_RUNNING_CHAINS" on page 5-31
- "USER_SCHEDULER_RUNNING_CHAINS" on page 5-127

ALL_SCHEDULER_RUNNING_JOBS

ALL_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs accessible to the current user.

Related Views

- DBA_SCHEDULER_RUNNING_JOBS displays information about all running Scheduler jobs in the database.
- USER_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the running Scheduler job
JOB_NAME	VARCHAR2(30)		Name of the running Scheduler job
JOB_SUBNAME	VARCHAR2(30)		Subname of the running Scheduler job (for a job running a chain step)

Column	Datatype	NULL	Description
JOB_STYLE	VARCHAR2(11)		Job style: ■ REGULAR ■ LIGHTWEIGHT
DETACHED	VARCHAR2(5)		Indicates whether the detached attribute is set for the job (TRUE) or not (FALSE). If the detached attribute is set, then the job will remain running even after the job action has completed.
SESSION_ID	NUMBER		Identifier of the session running the Scheduler job
SLAVE_PROCESS_ID	NUMBER		Process number of the slave process running the Scheduler job
SLAVE_OS_PROCESS_ID	VARCHAR2(12)		Process number of the operating system slave process running the scheduler job
RUNNING_INSTANCE	NUMBER		Database instance number of the slave process running the Scheduler job
RESOURCE_CONSUMER_GROUP	VARCHAR2(32)		Resource consumer group of the session in which the Scheduler job is running
ELAPSED_TIME	INTERVAL DAY(3) TO SECOND(2)		Elapsed time since the Scheduler job was started
CPU_USED	INTERVAL DAY(3) TO SECOND(2)		CPU time consumed by the running Scheduler job, if available

See Also:

- "DBA_SCHEDULER_RUNNING_JOBS" on page 5-31
- "USER_SCHEDULER_RUNNING_JOBS" on page 5-127

ALL_SCHEDULER_SCHEDULES

ALL_SCHEDULER_SCHEDULES displays information about the Scheduler schedules accessible to the current user.

Related Views

- DBA_SCHEDULER_SCHEDULES displays information about all Scheduler schedules in the database.
- USER_SCHEDULER_SCHEDULES displays information about the Scheduler schedules owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the schedule
SCHEDULE_NAME	VARCHAR2(30)	NOT NULL	Name of the schedule
SCHEDULE_TYPE	VARCHAR2(8)		Type of the schedule: ■ ONCE - Repeat interval is NULL ■ CALENDAR - Oracle calendaring expression used as schedule ■ EVENT - Event schedule
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Start date for the repeat interval
REPEAT_INTERVAL	VARCHAR2(4000)		Calendar syntax expression for the schedule
EVENT_QUEUE_OWNER	VARCHAR2(30)		Owner of the source queue into which the event will be raised

Column	Datatype	NULL	Description
EVENT_QUEUE_NAME	VARCHAR2(30)		Name of the source queue into which the event will be raised
EVENT_QUEUE_AGENT	VARCHAR2(30)		Name of the AQ agent used by the user on the event source queue (if it is a secure queue)
EVENT_CONDITION	VARCHAR2(4000)		Boolean expression used as the subscription rule for the event on the source queue
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Cutoff date after which the schedule will not specify any dates
COMMENTS	VARCHAR2(240)		Comments on the schedule

See Also:

- "DBA_SCHEDULER_SCHEDULES" on page 5-31
- "USER_SCHEDULER_SCHEDULES" on page 5-127

ALL_SCHEDULER_WINDOW_DETAILS

ALL_SCHEDULER_WINDOW_DETAILS displays log details for the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_DETAILS displays log details for all Scheduler windows in the database.

Column	Datatype	NULL	Description
LOG_ID	NUMBER		Unique identifier of the log entry (foreign key of the *_SCHEDULER_WINDOW_LOG views)
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
WINDOW_NAME	VARCHAR2(65)		Name of the Scheduler window
REQ_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Requested start date for the Scheduler window
ACTUAL_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Actual start date of the Scheduler window
WINDOW_DURATION	INTERVAL DAY(3) TO SECOND(0)		Requested duration of the Scheduler window
ACTUAL_DURATION	INTERVAL DAY(3) TO SECOND(0)		Actual duration for which the Scheduler window lasted
INSTANCE_ID	NUMBER		Identifier of the instance on which the window was run
ADDITIONAL_INFO	VARCHAR2(4000)		Additional information on the entry, if applicable

See Also: "DBA_SCHEDULER_WINDOW_DETAILS" on page 5-31

ALL_SCHEDULER_WINDOW_GROUPS

ALL_SCHEDULER_WINDOW_GROUPS displays information about the Scheduler window groups accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_GROUPS displays information about all Scheduler window groups in the database.

Column	Datatype	NULL	Description
WINDOW_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the window group
ENABLED	VARCHAR2(5)		Indicates whether the window group is enabled (TRUE) or disabled (FALSE)
NUMBER_OF_WINDOWS	NUMBER		Number of members in the window group
NEXT_START_DATE	VARCHAR2(64)		If a window group is disabled, then this column will be NULL. Otherwise, it will be set to the earliest NEXT_START_DATE from the enabled windows in the group.
COMMENTS	VARCHAR2(240)		Optional comment about the window group

See Also: "DBA_SCHEDULER_WINDOW_GROUPS" on page 5-31

ALL_SCHEDULER_WINDOW_LOG

ALL_SCHEDULER_WINDOW_LOG displays log information for the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOW_LOG displays log information for all Scheduler windows in the database.

Column	Datatype	NULL	Description
LOG_ID	NUMBER	NOT NULL	Unique identifier of the log entry
LOG_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the log entry
WINDOW_NAME	VARCHAR2(65)		Name of the Scheduler window
OPERATION	VARCHAR2(30)		Operation corresponding to the log entry
STATUS	VARCHAR2(30)		Status of the operation, if applicable
USER_NAME	VARCHAR2(30)		Name of the user who performed the operation, if applicable
CLIENT_ID	VARCHAR2(64)		Client identifier of the user who performed the operation, if applicable
GLOBAL_UID	VARCHAR2(32)		Global user identifier of the user who performed the operation, if applicable
ADDITIONAL_INFO	CLOB		Additional information on the entry, if applicable

See Also: "DBA_SCHEDULER_WINDOW_LOG" on page 5-32

ALL_SCHEDULER_WINDOWS

ALL_SCHEDULER_WINDOWS displays information about the Scheduler windows accessible to the current user.

Related View

DBA_SCHEDULER_WINDOWS displays information about all Scheduler windows in the database.

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler window
RESOURCE_PLAN	VARCHAR2(30)		Resource plan associated with the window
SCHEDULE_OWNER	VARCHAR2(4000)		Owner of the schedule of the window
SCHEDULE_NAME	VARCHAR2(4000)		Name of the schedule of the window
SCHEDULE_TYPE	VARCHAR2(8)		Type of the schedule of the window: ■ ONCE - Repeat interval is NULL ■ NAMED - Named schedule ■ CALENDAR - Oracle calendaring expression used as schedule
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Start date of the window (for an inline schedule)
REPEAT_INTERVAL	VARCHAR2(4000)		Calendar string for the window (for an inline schedule)
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Date after which the window will no longer open (for an inline schedule)
DURATION	INTERVAL DAY(3) TO SECOND(0)		Duration of the window
WINDOW_PRIORITY	VARCHAR2(4)		Priority of the job relative to other windows: ■ HIGH ■ LOW
NEXT_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Next date on which the window is scheduled to start
LAST_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Last date on which the window opened
ENABLED	VARCHAR2(5)		Indicates whether the window is enabled (TRUE) or disabled (FALSE)
ACTIVE	VARCHAR2(5)		Indicates whether the window is open (TRUE) or not (FALSE)
MANUAL_OPEN_TIME	TIMESTAMP(6) WITH TIME ZONE		Open time of the window if it was manually opened, else NULL
MANUAL_DURATION	INTERVAL DAY(3) TO SECOND(0)		Duration of the window if it was manually opened, else NULL
COMMENTS	VARCHAR2(240)		Comments on the window

See Also: "DBA_SCHEDULER_WINDOWS" on page 5-32

ALL_SCHEDULER_WINGROUP_MEMBERS

ALL_SCHEDULER_WINGROUP_MEMBERS displays the members of the Scheduler window groups accessible to the current user.

Related View

DBA_SCHEDULER_WINGROUP_MEMBERS displays the members of all Scheduler window groups in the database.

Column	Datatype	NULL	Description
WINDOW_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the window group
WINDOW_NAME	VARCHAR2(30)	NOT NULL	Name of the window member of the window group

See Also: "DBA_SCHEDULER_WINGROUP_MEMBERS" on page 5-32

ALL_SEC_RELEVANT_COLS

ALL_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views accessible to the current user.

Related Views

- DBA_SEC_RELEVANT_COLS describes the security relevant columns of all security policies in the database.
- USER_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views owned by the current user. This view does not display the OBJECT_OWNER column.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or view
OBJECT_NAME	VARCHAR2(30)		Name of the table or view
POLICY_GROUP	VARCHAR2(30)		Name of the policy group
POLICY_NAME	VARCHAR2(30)		Name of the policy
SEC_REL_COLUMN	VARCHAR2(30)		Name of the security relevant column
COLUMN_OPTION	VARCHAR2(8)		Option of the security relevant column: ■ NONE ■ ALL_ROWS

See Also:

- "DBA_SEC_RELEVANT_COLS" on page 5-32
- "USER_SEC_RELEVANT_COLS" on page 5-127

ALL_SECONDARY_OBJECTS

ALL_SECONDARY_OBJECTS provides information about secondary objects associated with domain indexes accessible to the user. This view is only relevant in the context of domain indexes. And currently, the secondary objects can only be tables.

Related Views

- DBA_SECONDARY_OBJECTS provides information about all secondary objects that are associated with domain indexes in the database.
- USER_SECONDARY_OBJECTS provides information about secondary objects associated with domain indexes owned by the current user.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Name of the domain index owner
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the domain index
SECONDARY_OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the secondary object created by the domain index
SECONDARY_OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the secondary object created by the domain index

Column	Datatype	NULL	Description
SECONDARY_OBJDATA_TYPE	VARCHAR2(20)		Type of the secondary object created by the domain index

ALL_SEQUENCES

ALL_SEQUENCES describes all sequences accessible to the current user.

Related Views

- DBA_SEQUENCES describes all sequences in the database.
- USER_SEQUENCES describes all sequences owned by the current user. This view does not display the SEQUENCE_OWNER column.

Column	Datatype	NULL	Description
SEQUENCE_OWNER	VARCHAR2(30)	NOT NULL	Name of the owner of the sequence
SEQUENCE_NAME	VARCHAR2(30)	NOT NULL	Sequence name
MIN_VALUE	NUMBER		Minimum value of the sequence
MAX_VALUE	NUMBER		Maximum value of the sequence
INCREMENT_BY	NUMBER	NOT NULL	Value by which sequence is incremented
CYCLE_FLAG	VARCHAR2(1)		Does sequence wrap around on reaching limit
ORDER_FLAG	VARCHAR2(1)		Are sequence numbers generated in order
CACHE_SIZE	NUMBER	NOT NULL	Number of sequence numbers to cache
LAST_NUMBER	NUMBER	NOT NULL	Last sequence number written to disk. If a sequence uses caching, the number written to disk is the last number placed in the sequence cache. This number is likely to be greater than the last sequence number that was used.

ALL_SERVICES

ALL_SERVICES displays all services in the database. The view excludes rows marked for deletion.

Related View

DBA_SERVICES displays all services in the database. The view excludes rows marked for deletion.

Column	Datatype	NULL	Description
SERVICE_ID	NUMBER		Unique ID for the service
NAME	VARCHAR2(64)		Service name for a queue
NAME_HASH	NUMBER		Hash of the short name for the service
NETWORK_NAME	VARCHAR2(512)		Network name used to connect to the service
CREATION_DATE	DATE		Date the service was created
CREATION_DATE_HASH	NUMBER		Hash of the creation date
FAILOVER_METHOD	VARCHAR2(64)		Failover method (BASIC or NONE) for the service
FAILOVER_TYPE	VARCHAR2(64)		Failover type (SESSION or SELECT) for the service
FAILOVER_RETRIES	NUMBER(10)		Number of retries when failing over the service
FAILOVER_DELAY	NUMBER(10)		Delay between retries when failing over the service

Column	Datatype	NULL	Description
MIN_CARDINALITY	NUMBER		Minimum cardinality of this service to be maintained by the director
MAX_CARDINALITY	NUMBER		Maximum cardinality of this service to be allowed by the director
GOAL	VARCHAR2(12)		Service workload management goal: ▪ NONE ▪ SERVICE_TIME ▪ THROUGHPUT
DTP	VARCHAR2(1)		DTP flag for services (Y) or (N)
ENABLED	VARCHAR2(3)		Indicates whether this service will be started/maintained by the director (YES) or not (NO)
AQ_HA_NOTIFICATIONS	VARCHAR2(3)		Indicates whether AQ notifications are sent for HA events (YES) or not (NO)
CLB_GOAL	VARCHAR2(5)		The connection load balancing goal. It is used with statistics that are sent to the listeners to determine how new connections are distributed. It may be set to either LONG or SHORT (the default).

ALL_SOURCE

ALL_SOURCE describes the text source of the stored objects accessible to the current user.

Related Views

- DBA_SOURCE describes the text source of all stored objects in the database.
- USER_SOURCE describes the text source of the stored objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of object: FUNCTION, JAVA SOURCE, PACKAGE, PACKAGE BODY, PROCEDURE, TRIGGER, TYPE, TYPE BODY
LINE	NUMBER	NOT NULL	Line number of this line of source
TEXT	VARCHAR2(4000)		Text source of the stored object

ALL_SOURCE_TABLES

ALL_SOURCE_TABLES is a synonym for USER_SOURCE_TABLES.

See Also: "USER_SOURCE_TABLES" on page 5-128

ALL_SQLJ_TYPE_ATTRS

ALL_SQLJ_TYPE_ATTRS describes the attributes of the SQLJ object types accessible to the current user.

Related Views

- DBA_SQLJ_TYPE_ATTRS describes the attributes of all SQLJ object types in the database.

- USER_SQLJ_TYPE_ATTRS describes the attributes of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
ATTR_NAME	VARCHAR2(30)	NOT NULL	Name of the attribute
EXTERNAL_ATTR_NAME	VARCHAR2(4000)		External name of the attribute
ATTR_TYPE_MOD	VARCHAR2(7)		Type modifier of the attribute: ■ REF ■ POINTER
ATTR_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the attribute
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute, or maximum length of the VARCHAR or VARCHAR2 attribute.
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute, or binary precision of the FLOAT attribute.
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET _NAME	VARCHAR2(44)		Character set name of the attribute (CHAR_CS or NCHAR_CS)
ATTR_NO	NUMBER	NOT NULL	Syntactical order number or position of the attribute as specified in the type specification or CREATE TYPE statement (not to be used as an ID number)
INHERITED	VARCHAR2(3)		Indicates whether the attribute is inherited from a supertype (YES) or not (NO)

See Also:

- "DBA_SQLJ_TYPE_ATTRS" on page 5-38
- "USER_SQLJ_TYPE_ATTRS" on page 5-128

ALL_SQLJ_TYPE_METHODS

ALL_SQLJ_TYPE_METHODS describes the methods of the SQLJ object types accessible to the current user.

Related Views

- DBA_SQLJ_TYPE_METHODS describes the methods of all SQLJ object types in the database.
- USER_SQLJ_TYPE_METHODS describes the methods of the SQLJ object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
EXTERNAL_VAR_NAME	VARCHAR2(4000)		Name of the external variable
METHOD_NO	NUMBER	NOT NULL	Method number that distinguishes overloaded methods (not to be used as an ID number)

Column	Datatype	NULL	Description
METHOD_TYPE	VARCHAR2(6)		Type of the method: ■ MAP ■ ORDER ■ PUBLIC
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method
FINAL	VARCHAR2(3)		Indicates whether the method is final (YES) or not (NO)
INSTANTIABLE	VARCHAR2(3)		Indicates whether the method is instantiable (YES) or not (NO)
OVERRIDING	VARCHAR2(3)		Indicates whether the method is overriding a supertype method (YES) or not (NO)
INHERITED	VARCHAR2(3)		Indicates whether the method is inherited from a supertype (YES) or not (NO)

See Also:

- "DBA_SQLJ_TYPE_METHODS" on page 5-38
- "USER_SQLJ_TYPE_METHODS" on page 5-128

ALL_SQLJ_TYPES

ALL_SQLJ_TYPES describes the SQLJ object types accessible to the current user.

Related Views

- DBA_SQLJ_TYPES describes all SQLJ object types in the database.
- USER_SQLJ_TYPES describes the SQLJ object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
TYPE_OID	RAW(16)	NOT NULL	Object identifier (OID) of the type
EXTERNAL_NAME	VARCHAR2(4000)		External class name of the type
USING	VARCHAR2(21)		Representation of the type: ■ SQLData ■ CustomDatum ■ Serializable ■ Serializable Internal ■ ORAData
TYPECODE	VARCHAR2(30)		Typecode of the type
ATTRIBUTES	NUMBER		Number of attributes (if any) in the type
METHODS	NUMBER		Number of methods (if any) in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type (YES) or not (NO)
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type (YES) or not (NO)
FINAL	VARCHAR2(3)		Indicates whether the type is a final type (YES) or not (NO)

Column	Datatype	NULL	Description
INSTANTIABLE	VARCHAR2(3)		Indicates whether the type is an instantiable type (YES) or not (NO)
SUPERTYPE_OWNER	VARCHAR2(30)		Owner of the supertype (null if type is not a subtype)
SUPERTYPE_NAME	VARCHAR2(30)		Name of the supertype (null if type is not a subtype)
LOCAL_ATTRIBUTES	NUMBER		Number of local (not inherited) attributes (if any) in the subtype
LOCAL_METHODS	NUMBER		Number of local (not inherited) methods (if any) in the subtype

See Also:

- "DBA_SQLJ_TYPES" on page 5-39
- "USER_SQLJ_TYPES" on page 5-128

ALL_SQLSET

ALL_SQLSET displays information about all SQL tuning sets accessible to the current user.

Related Views

- DBA_SQLSET displays information about all SQL tuning sets in the database.
- USER_SQLSET displays information about the SQL tuning sets owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the SQL tuning set
ID	NUMBER	NOT NULL	SQL tuning set identifier
OWNER	VARCHAR2(30)		Owner of the SQL tuning set
DESCRIPTION	VARCHAR2(256)		Description of the SQL tuning set
CREATED	DATE		Date the SQL tuning set was created
LAST_MODIFIED	DATE		Date the SQL tuning set was last modified
STATEMENT_COUNT	NUMBER		Number of statements in the SQL tuning set

ALL_SQLSET_BINDS

ALL_SQLSET_BINDS displays the bind values associated with all SQL tuning sets accessible to the current user.

Related Views

- DBA_SQLSET_BINDS displays the bind values associated with all SQL tuning sets in the database.
- USER_SQLSET_BINDS displays the bind values associated with the SQL tuning sets owned by the current user. This view does not display the SQLSET_OWNER column.

Column	Datatype	NULL	Description
SQLSET_NAME	VARCHAR2(30)	NOT NULL	Name of the SQL tuning set for the statement
SQLSET_OWNER	VARCHAR2(30)		User name of the SQL tuning set owner

Column	Datatype	NULL	Description
SQLSET_ID	NUMBER	NOT NULL	ID of the SQL tuning set for the statement
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
FORCE_MATCHING_SIGNATURE	NUMBER	NOT NULL	The signature used when the CURSOR_SHARING parameter is set to FORCE
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line-by-line).
POSITION	NUMBER	NOT NULL	Position
VALUE	ANYDATA		Value
CAPTURED	CHAR(1)		Binds captured
SQL_SEQ	NUMBER	NOT NULL	SQL sequence

ALL_SQLSET_PLANS

ALL_SQLSET_PLANS describes captured plans for statements in the SQL tuning sets accessible to the current user.

Related Views

- DBA_SQLSET_PLANS describes captured plans in the SQL tuning sets in the database.
- USER_SQLSET_PLANS describes captured plans for statements in the SQL tuning sets owned by the current user. This view does not display the SQLSET_OWNER column.

Column	Datatype	NULL	Description
SQLSET_NAME	VARCHAR2(30)	NOT NULL	Name of SQL tuning set for the statement
SQLSET_ID	NUMBER	NOT NULL	ID of SQL tuning set for the statement
SQLSET_OWNER	VARCHAR2(30)		User name of SQL tuning set owner
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
FORCE_MATCHING_SIGNATURE	NUMBER	NOT NULL	The signature used when the CURSOR_SHARING parameter is set to FORCE
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line-by-line).
SQL_SEQ	NUMBER	NOT NULL	SQL sequence
STATEMENT_ID	VARCHAR2(30)		Statement ID
PLAN_ID	NUMBER		Plan ID
TIMESTAMP	DATE		Data and time timestamp
REMARKS	VARCHAR2(4000)		Remarks
OPERATION	VARCHAR2(30)		Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(255)		A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(30)		Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)		Alias for the object
OBJECT_INSTANCE	NUMBER(38)		Instance number for the object
OBJECT_TYPE	VARCHAR2(30)		Type of the object
OPTIMIZER	VARCHAR2(255)		Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
ID	NUMBER(38)	NOT NULL	A number assigned to each step in the execution plan
PARENT_ID	NUMBER(38)		ID of the next execution step that operates on the output of the current step
DEPTH	NUMBER(38)		Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER(38)		Order of processing for all operations that have the same PARENT_ID.
COST	NUMBER(38)		Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER(38)		Estimate, made by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER(38)		Estimate, made by the cost-based optimizer, of the number of bytes produced by the operation
OTHER_TAG	VARCHAR2(255)		Describes the contents of the OTHER column. For information about values, see EXPLAIN PLAN in *Oracle Database Performance Tuning Guide*.
PARTITION_START	VARCHAR2(255)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(255)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER(38)		Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Other information specific to the execution step that users may find useful. For information about values, see EXPLAIN_PLAN in *Oracle Database Performance Tuning Guide*.
DISTRIBUTION	VARCHAR2(30)		Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER(38)		CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER(38)		I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER(38)		Temporary space usage of the operation (sort or hash join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.

Column	Datatype	NULL	Description
FILTER_PREDICATES	VARCHAR2(4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)		Expressions produced by the operation
TIME	NUMBER(38)		Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(30)		Name of the query block
OTHER_XML	CLOB		Provides extra information specific to an execution step of the execution plan. The content of this column is structured using XML since multiple pieces of information can be stored there. This includes: ■ Name of the schema against which the query was parsed ■ Release number of the Oracle Database that produced the explain plan ■ Hash value associated with the execution plan ■ Name (if any) of the outline or the SQL profile used to build the execution plan ■ Indication of whether or not dynamic sampling was used to produce the plan ■ The outline data, a set of optimizer hints that can be used to regenerate the same plan For further information about values, see EXPLAIN PLAN in *Oracle Database Performance Tuning Guide*
EXECUTIONS	NUMBER		Number of times the plan has been executed
STARTS	NUMBER		Number of times this operation has been started, accumulated over the past executions
OUTPUT_ROWS	NUMBER		Number of rows produced by the row source, accumulated over the past executions
CR_BUFFER_GETS	NUMBER		Number of buffers received in consistent mode, accumulated over the past executions. Buffers are usually retrieved in consistent mode for queries.
CU_BUFFER_GETS	NUMBER		Number of buffers retrieved in current mode, accumulated over the past executions. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
DISK_READS	NUMBER		Number of physical disk reads performed by the operation, accumulated over the past executions
DISK_WRITES	NUMBER		Number of physical disk writes performed by the operation, accumulated over the past executions
ELAPSED_TIME	NUMBER		Elapsed time (in microseconds) corresponding to this operation, accumulated over the past executions
LAST_STARTS	NUMBER		???
LAST_OUTPUT_ROWS	NUMBER		???
LAST_CR_BUFFER_GETS	NUMBER		???
LAST_CU_BUFFER_GETS	NUMBER		???
LAST_DISK_READS	NUMBER		???
LAST_DISK_WRITES	NUMBER		???
LAST_ELAPSED_TIME	NUMBER		???
POLICY	VARCHAR2(10)		???
ESTIMATED_OPTIMAL_SIZE	NUMBER		???
ESTIMATED_ONEPASS_SIZE	NUMBER		???

Column	Datatype	NULL	Description
LAST_MEMORY_USED	NUMBER		???
LAST_EXECUTION	VARCHAR2(10)		???
LAST_DEGREE	NUMBER		???
TOTAL_EXECUTIONS	NUMBER		???
OPTIMAL_EXECUTIONS	NUMBER		???
ONEPASS_EXECUTIONS	NUMBER		???
MULTIPASSES_ EXECUTIONS	NUMBER		???
ACTIVE_TIME	NUMBER		???
MAX_TEMPSEG_SIZE	NUMBER		???
LAST_TEMPSEG_SIZE	NUMBER		???

ALL_SQLSET_REFERENCES

ALL_SQLSET_REFERENCES describes whether or not the SQL tuning sets accessible to the current user are active.

Related Views

- DBA_SQLSET_REFERENCES describes whether or not all SQL tuning sets in the database are active. A SQL tuning set cannot be dropped if it is referenced.
- USER_SQLSET_REFERENCES describes whether or not the SQL tuning sets owned by the current user are active.

Column	Datatype	NULL	Description
SQLSET_NAME	VARCHAR2(30)	NOT NULL	Name of the SQL tuning set
SQLSET_ID	NUMBER	NOT NULL	Identifier of the SQL tuning set
SQLSET_OWNER	VARCHAR2(30)		Owner of the SQL tuning set
ID	NUMBER	NOT NULL	Reference identifier
OWNER	VARCHAR2(30)		User who registered to use the SQL tuning set
DESCRIPTION	VARCHAR2(256)		Description of the usage of the SQL tuning set
CREATED	DATE		Date the reference was created

ALL_SQLSET_STATEMENTS

ALL_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form all SQL tuning sets accessible to the current user.

Related Views

- DBA_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form all SQL tuning sets in the database. This view does not display the SQL_SEQ column.
- USER_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form the SQL tuning sets owned by the current user. This view does not display the SQLSET_OWNER column.

Column	Datatype	NULL	Description
SQLSET_NAME	VARCHAR2(30)	NOT NULL	Name of the SQL tuning set for the statement

Column	Datatype	NULL	Description
SQLSET_ID	NUMBER	NOT NULL	ID of the SQL tuning set for the statement
SQLSET_OWNER	VARCHAR2(30)		User name of the SQL tuning set owner
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
FORCE_MATCHING_SIGNATURE	NUMBER	NOT NULL	The signature used when the CURSOR_SHARING parameter is set to FORCE
SQL_TEXT	CLOB		Full text for the SQL statement exposed as a CLOB column.
PLAN_HASH_VALUE	NUMBER	NOT NULL	Hash value for the plan corresponding to statistics in this record
PARSING_SCHEMA_NAME	VARCHAR2(30)		Name of the user in whose schema the statement was parsed
BIND_DATA	RAW(2000)		Bind data
BINDS_CAPTURED	CHAR(1)		Binds captured
MODULE	VARCHAR2(48)		Contains the name of the module that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_MODULE
ACTION	VARCHAR2(32)		Contains the name of the action that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_ACTION
ELAPSED_TIME	NUMBER		Elapsed time (in microseconds) used by this cursor for parsing, executing, and fetching
CPU_TIME	NUMBER		CPU time (in microseconds) used by this cursor for parsing, executing, and fetching
BUFFER_GETS	NUMBER		Number of buffer gets for this child cursor
DISK_READS	NUMBER		Number of disk reads for this child cursor
DIRECT_WRITES	NUMBER		Number of direct writes for this child cursor
ROWS_PROCESSED	NUMBER		Total number of rows that the parsed SQL statement returns
FETCHES	NUMBER		Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER		Number of executions that took place on this object since it was brought into the library cache
END_OF_FETCH_COUNT	NUMBER		Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic in not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less than, or equal to, the value of the EXECUTIONS column.
OPTIMIZER_COST	NUMBER		Cost of this query, given by the optimizer
OPTIMIZER_ENV	RAW(1000)		Optimizer environment
PRIORITY	NUMBER		User-defined priority
COMMAND_TYPE	NUMBER		Oracle command type definition
FIRST_LOAD_TIME	VARCHAR2(19)		Timestamp of the parent creation time
STAT_PERIOD	NUMBER		Period of time (in seconds) during which the statistics of the SQL statement were collected
ACTIVE_STAT_PERIOD	NUMBER		Effective period of time (in seconds) during which the SQL statement was active
OTHER	CLOB		Client data, specified by the user, for this statement

Column	Datatype	NULL	Description
PLAN_TIMESTAMP	DATE		Timestamp for the plan corresponding to the statistics in this record
SQL_SEQ	NUMBER	NOT NULL	SQL sequence

ALL_STAT_EXTENSIONS

ALL_STAT_EXTENSIONS displays information about the optimizer statistics extensions accessible to the current user.

Related Views

- DBA_STAT_EXTENSIONS displays information about all optimizer statistics extensions in the database.
- USER_STAT_EXTENSIONS displays information about the optimizer statistics extensions owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the extension
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table to which the extension belongs
EXTENSION_NAME	VARCHAR2(30)	NOT NULL	Name of the statistics extension
EXTENSION	CLOB		Extension (the expression or column group)
CREATOR	VARCHAR2(6)		Creator of the extension: ■ USER ■ SYSTEM
DROPPABLE	VARCHAR2(3)		Indicates whether the extension is droppable using DBMS_STATS.DROP_EXTENDED_STATS (YES) or not (NO)

See Also:

- "DBA_STAT_EXTENSIONS" on page 5-43
- "USER_STAT_EXTENSIONS" on page 5-129

ALL_STORED_SETTINGS

ALL_STORED_SETTINGS provides information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges.

Related Views

- DBA_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges. It also returns parameter information for all objects in the database and is accessible only to users with the SELECT_CATALOG_ROLE privilege.
- USER_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units, but only shows information about PL/SQL units owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the database user owning the stored PL/SQL unit
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the PL/SQL unit
OBJECT_ID	NUMBER	NOT NULL	Object number of the PL/SQL unit
OBJECT_TYPE	VARCHAR2(12)		The type of PL/SQL unit: PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, TYPE, or TYPE BODY
PARAM_NAME	VARCHAR2(30)		The name of the parameter stored persistently with the PL/SQL unit
PARAM_VALUE	VARCHAR2(4000)		The TO_CHAR() representation of the value of the persistently stored parameter. The width of this column is operating system dependent; however, it is never less than 255.

Note: This view is deprecated in favor of the ALL_PLSQL_OBJECT_SETTINGS view. Oracle recommends that you use ALL_PLSQL_OBJECT_SETTINGS instead. ALL_STORED_SETTINGSis retained for backward compatibility only.

ALL_STREAMS_COLUMNS

ALL_STREAMS_COLUMNS displays information about the Streams unsupported columns accessible to the current user.

Related View

DBA_STREAMS_COLUMNS displays information about all Streams unsupported columns in the database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(30)		Name of the column
SYNC_CAPTURE_VERSION	NUMBER		Version of the synchronous capture that supports this column
SYNC_CAPTURE_REASON	VARCHAR2(39)		Reason why this column is not supported by synchronous capture
APPLY_VERSION	NUMBER		Version of apply that supports this column
APPLY_REASON	VARCHAR2(40)		Reason why this column is not supported by apply

See Also: "DBA_STREAMS_COLUMNS" on page 5-45

ALL_STREAMS_GLOBAL_RULES

ALL_STREAMS_GLOBAL_RULES displays information about the following types of rules:

- Global rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Global rules created for the Streams propagations that have a source queue accessible to the current user

- Global rules created for the Streams apply processes that dequeue events from queues accessible to the current user

This view only contains information about rules created using the ADD_GLOBAL_RULES or ADD_GLOBAL_PROPAGATION_RULES procedures in the DBMS_STREAMS_ADM package. It does not contain information about rules created using the DBMS_RULE_ADM package.

Related View

DBA_STREAMS_GLOBAL_RULES displays information about the global rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process or propagation: ■ CAPTURE ■ PROPAGATION ■ APPLY
RULE_TYPE	VARCHAR2(7)		Type of the rule: ■ DML ■ DDL
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-NULL tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_GLOBAL_RULES" on page 5-45

ALL_STREAMS_MESSAGE_CONSUMERS

ALL_STREAMS_MESSAGE_CONSUMERS displays information about the Streams messaging clients accessible to the current user.

Related View

DBA_STREAMS_MESSAGE_CONSUMERS displays information about all Streams messaging clients in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)	NOT NULL	Name of the messaging client
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue
RULE_SET_NAME	VARCHAR2(30)		Name of the positive rule set
RULE_SET_OWNER	VARCHAR2(30)		Owner of the positive rule set

Column	Datatype	NULL	Description
NEGATIVE_RULE_SET_NAME	VARCHAR2(30)		Name of the negative rule set
NEGATIVE_RULE_SET_OWNER	VARCHAR2(30)		Owner of the negative rule set
NOTIFICATION_TYPE	VARCHAR2(9)		Type of the notification action: ■ PROCEDURE ■ MAIL ■ HTTP
NOTIFICATION_ACTION	VARCHAR2(256)		Notification action
NOTIFICATION_CONTEXT	ANYDATA		Context for the notification action

See Also: "DBA_STREAMS_MESSAGE_CONSUMERS" on page 5-45

ALL_STREAMS_MESSAGE_RULES

ALL_STREAMS_MESSAGE_RULES displays information about the Streams messaging rules accessible to the current user.

Related View

DBA_STREAMS_MESSAGE_RULES displays information about all Streams messaging rules in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams client
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams client: ■ PROPAGATION - Streams propagation ■ APPLY - Streams apply process ■ DEQUEUE - Streams messaging client
MESSAGE_TYPE_NAME	VARCHAR2(30)		Name of the message type
MESSAGE_TYPE_OWNER	VARCHAR2(30)		Owner of the message type
MESSAGE_RULE_VARIABLE	VARCHAR2(30)		Name of the variable in the message rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		Rule condition

See Also: "DBA_STREAMS_MESSAGE_RULES" on page 5-46

ALL_STREAMS_NEWLY_SUPPORTED

ALL_STREAMS_NEWLY_SUPPORTED displays information about the tables accessible to the current user that are newly supported by Streams.

Related View

DBA_NEWLY_SUPPORTED displays information about all tables in the database that are newly supported by Streams.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
REASON	VARCHAR2(39)		Reason why the table was not supported in a previous release: - IOT - column with user-defined type - unsupported column exists - object table - AQ queue table - temporary table - sub object - external table - materialized view - FILE column exists - materialized view log - materialized view container table - streams unsupported object - domain index
COMPATIBLE	CHAR(4)		Latest COMPATIBLE setting when the table was unsupported

See Also: "DBA_STREAMS_NEWLY_SUPPORTED" on page 5-46

ALL_STREAMS_RULES

ALL_STREAMS_RULES displays information about the rules used by the Streams processes accessible to the current user.

Related View

DBA_STREAMS_RULES displays information about the rules used by all Streams processes in the database.

Column	Datatype	NULL	Description
STREAMS_TYPE	VARCHAR2(12)		Type of the Streams process: - CAPTURE - PROPAGATION - APPLY - DEQUEUE - SYNC_CAPTURE
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set
RULE_SET_NAME	VARCHAR2(30)		Name of the rule set
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_CONDITION	CLOB		Current rule condition
RULE_SET_TYPE	CHAR(8)		Type of the rule set: - POSITIVE - NEGATIVE

Column	Datatype	NULL	Description
STREAMS_RULE_TYPE	VARCHAR2(6)		For global, schema, or table rules, the type of the rule: ■ TABLE ■ SCHEMA ■ GLOBAL
SCHEMA_NAME	VARCHAR2(30)		For table and schema rules, the schema name
OBJECT_NAME	VARCHAR2(30)		For table rules, the table name
SUBSETTING_OPERATION	VARCHAR2(6)		For subset rules, the type of operation: ■ INSERT ■ UPDATE ■ DELETE
DML_CONDITION	VARCHAR2(4000)		For subset rules, the row subsetting condition
INCLUDE_TAGGED_LCR	VARCHAR2(3)		For global, schema, or table rules, indicates whether to include tagged logical change records (LCRs) (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		For global, schema, or table rules, the name of the database where the logical change records (LCRs) originated
RULE_TYPE	VARCHAR2(3)		For global, schema, or table rules, the type of the rule: ■ DML ■ DDL
MESSAGE_TYPE_OWNER	VARCHAR2(30)		For message rules, the owner of the message type
MESSAGE_TYPE_NAME	VARCHAR2(30)		For message rules, the name of the message type
MESSAGE_RULE_VARIABLE	VARCHAR2(30)		For message rules, the name of the variable in the message rule
ORIGINAL_RULE_CONDITION	VARCHAR2(4000)		For rules created by Streams administrative APIs, the original rule condition when the rule was created
SAME_RULE_CONDITION	VARCHAR2(3)		For rules created by Streams administrative APIs, indicates whether the current rule condition is the same as the original rule condition (YES) or not (NO)

See Also: "DBA_STREAMS_RULES" on page 5-47

ALL_STREAMS_SCHEMA_RULES

ALL_STREAMS_SCHEMA_RULES displays information about the following types of rules:

- Schema rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Schema rules created for the Streams propagations that have a source queue accessible to the current user
- Schema rules created for the Streams apply processes that dequeue events from queues accessible to the current user

This view only contains information about rules created using the ADD_SCHEMA_RULES or ADD_SCHEMA_PROPAGATION_RULES procedures in the DBMS_STREAMS_ADM package. It does not contain information about rules created using the DBMS_RULE_ADM package.

Related View

DBA_STREAMS_SCHEMA_RULES displays information about the schema rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(11)		Type of the Streams process or propagation: ■ CAPTURE ■ PROPAGATION ■ APPLY
SCHEMA_NAME	VARCHAR2(30)		Schema name in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this schema name.
RULE_TYPE	VARCHAR2(7)		Type of the rule: ■ DML ■ DDL
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-NULL tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_SCHEMA_RULES" on page 5-47

ALL_STREAMS_TABLE_RULES

ALL_STREAMS_TABLE_RULES displays information about the following types of rules:

- Table rules created for the Streams capture processes that enqueue the captured changes into queues accessible to the current user
- Table rules created for the Streams propagations that have a source queue accessible to the current user
- Table rules created for the Streams apply processes that dequeue events from queues accessible to the current user
- Subset rules created for the Streams apply processes that have a source queue accessible to the current user

This view only contains information about rules created using the ADD_TABLE_RULES, ADD_TABLE_PROPAGATION_RULES, or ADD_SUBSET_RULES procedures in the DBMS_STREAMS_ADM package. It does not contain information about rules created using the DBMS_RULE_ADM package.

Related View

DBA_STREAMS_TABLE_RULES displays information about the table rules created for all Streams capture processes, propagations, and apply processes in the database.

Column	Datatype	NULL	Description
STREAMS_NAME	VARCHAR2(30)		Name of the Streams process or propagation
STREAMS_TYPE	VARCHAR2(12)		Type of the Streams process or propagation: ■ CAPTURE ■ PROPAGATION ■ APPLY ■ DEQUEUE ■ SYNC_CAPTURE
TABLE_OWNER	VARCHAR2(30)		Table owner in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this table owner.
TABLE_NAME	VARCHAR2(30)		Table name in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this table name.
RULE_TYPE	VARCHAR2(7)		Type of the rule: ■ DML ■ DDL
DML_CONDITION	VARCHAR2(4000)		Row subsetting condition, if the rule is a subset rule
SUBSETTING_OPERATION	VARCHAR2(6)		DML operation for row subsetting in the rule condition, if the rule is a subset rule: ■ INSERT ■ UPDATE ■ DELETE The rule evaluates to true for a logical change record (LCR) only if the LCR contains this command type after internal transformation.
INCLUDE_TAGGED_LCR	VARCHAR2(3)		Indicates whether a redo entry or logical change record (LCR) with a non-null tag is considered for capture, propagation, or apply (YES) or not (NO)
SOURCE_DATABASE	VARCHAR2(128)		Source database in the rule condition. The rule evaluates to true for a redo entry or logical change record (LCR) only if the redo entry or LCR contains this source database.
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule
RULE_CONDITION	VARCHAR2(4000)		First 4000 bytes of the system-generated rule condition evaluated by the rules engine

See Also: "DBA_STREAMS_TABLE_RULES" on page 5-47

ALL_STREAMS_TRANSFORM_FUNCTION

ALL_STREAMS_TRANSFORM_FUNCTION displays information about the rule-based transformation functions accessible to the current user.

Related View

DBA_STREAMS_TRANSFORM_FUNCTION displays information about all rule-based transformation functions in the database.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule associated with the transformation function

Column	Datatype	NULL	Description
RULE_NAME	VARCHAR2(30)	NOT NULL	Name of the rule associated with the transformation function
VALUE_TYPE	VARCHAR2(4000)		Type of the transformation function name. This type must be VARCHAR2 for a rule-based transformation to work properly.
TRANSFORM_FUNCTION_NAME	VARCHAR2(4000)		Name of the transformation function (null if VALUE_TYPE is not VARCHAR2)
CUSTOM_TYPE	VARCHAR2(11)		Type of the transformation function: ▪ ONE TO ONE - One-to-one transformations ▪ ONE TO MANY - One-to-many transformations

See Also:

- "DBA_STREAMS_TRANSFORM_FUNCTION" on page 5-50
- *Oracle Streams Concepts and Administration* for more information about custom rule-based transformation

ALL_STREAMS_UNSUPPORTED

ALL_STREAMS_UNSUPPORTED displays information about the tables accessible to the current user that are not supported by Streams in this release of the Oracle Database.

Related View

DBA_STREAMS_UNSUPPORTED displays information about all tables in the database that are not supported by Streams in this release of the Oracle Database.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table

Column	Datatype	NULL	Description
REASON	VARCHAR2(39)		Reason why the table is not supported: ■ IOT ■ column with user-defined type ■ unsupported column exists ■ object table ■ AQ queue table ■ temporary table ■ sub object ■ external table ■ materialized view ■ FILE column exists ■ materialized view log ■ materialized view container table ■ streams unsupported object ■ domain index ■ IOT with overflow ■ IOT with LOB ■ IOT with physical Rowid mapping ■ mapping table for physical rowid of IOT ■ IOT with LOB ■ IOT with row movement ■ summary container table
AUTO_FILTERED	VARCHAR2(3)		Indicates whether Streams automatically filters out the object (YES) or not (NO)

See Also: "DBA_STREAMS_UNSUPPORTED" on page 5-51

ALL_SUBPART_COL_STATISTICS

ALL_SUBPART_COL_STATISTICS describes column statistics and histogram information for subpartitions of partitioned objects accessible to the current user.

Related Views

- DBA_SUBPART_COL_STATISTICS provides this information for all subpartitions in the database.
- USER_SUBPART_COL_STATISTICS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner name
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(4000)		Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column

Column	Datatype	NULL	Description
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics gathered on partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

ALL_SUBPART_HISTOGRAMS

ALL_SUBPART_HISTOGRAMS lists the actual histogram data (end-points per histogram) for histograms on table subpartitions accessible to the current user.

Related Views

- DBA_SUBPART_HISTOGRAMS provides this information for all subpartitions in the database.
- USER_SUBPART_HISTOGRAMS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

> **Note:** These views are populated only if you collect statistics on the index using the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner name
TABLE_NAME	VARCHAR2(30)		Table name
SUBPARTITION_NAME	VARCHAR2(30)		Table subpartition name
COLUMN_NAME	VARCHAR2(30)		Column name
BUCKET_NUMBER	NUMBER		Bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint values for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket

ALL_SUBPART_KEY_COLUMNS

ALL_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) accessible to the current user.

Related Views

- DBA_SUBPART_KEY_COLUMNS provides this information for all subpartitions in the database.
- USER_SUBPART_KEY_COLUMNS provides this information for subpartitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		The partitioned table or index owner
NAME	VARCHAR2(30)		The partitioned table or index name
OBJECT_TYPE	VARCHAR2(11)		The object type (TABLE or INDEX)
COLUMN_NAME	VARCHAR2(30)		The column name
COLUMN_POSITION	NUMBER		The position of the column within the subpartitioning key

See Also:

- "DBA_SUBPART_KEY_COLUMNS" on page 5-52
- "USER_SUBPART_KEY_COLUMNS" on page 5-130

ALL_SUBPARTITION_TEMPLATES

ALL_SUBPARTITION_TEMPLATES describes the subpartition templates accessible to the current user.

Related Views

- DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database.
- USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
SUBPARTITION_NAME	VARCHAR2(34)	NOT NULL	Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name of the subpartition
HIGH_BOUND	LONG		Literal list values of the subpartition

See Also:

- "DBA_SUBPARTITION_TEMPLATES" on page 5-52
- "USER_SUBPARTITION_TEMPLATES" on page 5-130

ALL_SUBSCRIBED_COLUMNS

ALL_SUBSCRIBED_COLUMNS is a synonym for USER_SUBSCRIBED_COLUMNS.

See Also: "USER_SUBSCRIBED_COLUMNS" on page 5-130

ALL_SUBSCRIBED_TABLES

ALL_SUBSCRIBED_TABLES is a synonym for USER_SUBSCRIBED_TABLES.

See Also: "USER_SUBSCRIBED_TABLES" on page 5-131

ALL_SUBSCRIPTIONS

ALL_SUBSCRIPTIONS is a synonym for USER_SUBSCRIPTIONS.

See Also: "USER_SUBSCRIPTIONS" on page 5-131

ALL_SUMDELTA

ALL_SUMDELTA lists direct path load entries accessible to the current user.

Column	Datatype	NULL	Description
TABLEOBJ#	NUMBER	NOT NULL	Object number of the table
PARTITIONOBJ#	NUMBER	NOT NULL	Object number of table partitions (if the table is partitioned)
DMLOPERATION	VARCHAR2(1)		Type of DML operation applied to the table
SCN	NUMBER	NOT NULL	SCN when the bulk DML occurred
TIMESTAMP	DATE	NOT NULL	Timestamp of log entry
LOWROWID	ROWID	NOT NULL	The start ROWID in the loaded rowid range
HIGHROWID	ROWID	NOT NULL	The end ROWID in the loaded rowid range

ALL_SYNC_CAPTURE

ALL_SYNC_CAPTURE displays information about the synchronous capture processes that store the captured changes in queues accessible to the current user.

Related View

DBA_SYNC_CAPTURE displays information about all synchronous capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue used for holding captured changes
QUEUE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the queue used for holding captured changes
RULE_SET_NAME	VARCHAR2(30)		Rule set used by the capture process
RULE_SET_OWNER	VARCHAR2(30)		Owner of the rule set
CAPTURE_USER	VARCHAR2(30)		Current user who is enqueueing captured messages

See Also: "DBA_SYNC_CAPTURE" on page 5-55

ALL_SYNC_CAPTURE_PREPARED_TABS

ALL_SYNC_CAPTURE_PREPARED_TABS displays information about the tables accessible to the current user that are prepared for synchronous capture instantiation.

Related View

DBA_SYNC_CAPTURE_PREPARED_TABS displays information about all tables in the database that are prepared for synchronous capture instantiation.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the table prepared for synchronous capture instantiation
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table prepared for synchronous capture instantiation
SCN	NUMBER	NOT NULL	SCN from which changes can be captured
TIMESTAMP	DATE		Time at which the table was ready to be instantiated

See Also: "DBA_SYNC_CAPTURE_PREPARED_TABS" on page 5-55

ALL_SYNC_CAPTURE_TABLES

ALL_SYNC_CAPTURE_TABLES displays information about the tables accessible to the current user that are captured by synchronous Streams captures.

Related View

DBA_SYNC_CAPTURE_TABLES displays information about all tables in the database that are captured by synchronous Streams captures.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the synchronous capture table
TABLE_NAME	VARCHAR2(30)		Name of the synchronous capture table
ENABLED	VARCHAR2(3)		Indicates whether synchronous Streams capture is enabled for the table (YES) or not (NO)

See Also: "DBA_SYNC_CAPTURE_TABLES" on page 5-55

ALL_SYNONYMS

ALL_SYNONYMS describes the synonyms accessible to the current user. The following criteria determine the list of synonyms that ALL_SYNONYMS shows:

- All private synonyms owned by the logged-in user, even if the base object pointed to is not accessible.
- All public synonyms, even if the base object pointed to is not accessible.
- All private synonyms owned by a different user, where the ultimate base object pointed to by that synonym or by any chain of nested synonyms, is know to be accessible because of a grant to the logged-in user, or a grant to a role in effect for this session.
- If the current session has any of the following privileges, then all synonyms that point directly to local objects are shown because it is assumed that the session can access those objects:
  - LOCK ANY TABLE
  - SELECT ANY TABLE
  - INSERT ANY TABLE

- UPDATE ANY TABLE
- DELETE ANY TABLE

Synonyms that point to remote objects are excluded because the system privileges just listed do not automatically convey access to those remote objects. Also, if the synonyms point to objects other than tables and views (such as sequences, PL/SQL procedures, and so on) then this rule may show synonyms that ultimately resolve to objects that this session cannot access.

- All private synonyms owned by a different user, where the synonym is via a database link, are excluded.

Related Views

- DBA_SYNONYMS describes all synonyms in the database.
- USER_SYNONYMS describes the synonyms owned by the current user. The USER_SYNONYMS view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the synonym
SYNONYM_NAME	VARCHAR2(30)	NOT NULL	Name of the synonym
TABLE_OWNER	VARCHAR2(30)		Owner of the object referenced by the synonym, or creator of the referring synonym if the target is a public synonym (that is, the object referred to by TABLE_NAME). Although the column is called TABLE_OWNER, the object owned is not necessarily a table. It can be any general object such as a view, sequence, stored procedure, synonym, and so on.
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object referenced by the synonym. Although the column is called TABLE_NAME, the object does not necessarily have to be a table. It can be any general object such as a view, sequence, stored procedure, synonym, and so on.
DB_LINK	VARCHAR2(128)		Name of the database link referenced, if any

See Also:

- "DBA_SYNONYMS" on page 5-55
- "USER_SYNONYMS" on page 5-131

ALL_TAB_COL_STATISTICS

ALL_TAB_COL_STATISTICS provides column statistics and histogram information extracted from "ALL_TAB_COLUMNS" on page 3-68.

Related Views

- DBA_TAB_COL_STATISTICS provides such information extracted from "DBA_TAB_COLUMNS" on page 5-56.
- USER_TAB_COL_STATISTICS provides such information extracted from "USER_TAB_COLUMNS" on page 5-131.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Table owner

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Table name
COLUMN_NAME	VARCHAR2(30)		Column name
NUM_DISTINCT	NUMBER		Number of distinct values in the column
LOW_VALUE	RAW(32)		Low value in the column
HIGH_VALUE	RAW(32)		High value in the column
DENSITY	NUMBER		Density of the column
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in histogram for the column
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

ALL_TAB_COLS

ALL_TAB_COLS describes the columns of the tables, views, and clusters accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

This view differs from "ALL_TAB_COLUMNS" on page 3-68 in that hidden columns are not filtered out.

Related Views

- DBA_TAB_COLS describes the columns of all tables, views, and clusters in the database.
- USER_TAB_COLS describes the columns of the tables, views, and clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table, view, or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table, view, or cluster
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes

Column	Datatype	NULL	Description
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY.
COLUMN_ID	NUMBER		Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column[1]
LOW_VALUE	RAW(32)		Low value in the column[1]
HIGH_VALUE	RAW(32)		High value in the column[1]
DENSITY	NUMBER		Density of the column[1]
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column **Note:** The number of buckets in a histogram is specified in the SIZE parameter of the SQL statement ANALYZE. However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2(1)		B | C. B indicates that the column uses BYTE length semantics. C indicates that the column uses CHAR length semantics. NULL indicates the datatype is not any of the following: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2(3)		Indicates whether the column data is in release 8.0 image format (YES) or not (NO)
DATA_UPGRADED	VARCHAR2(3)		Indicates whether the column data has been upgraded to the latest type version format (YES) or not (NO)

Column	Datatype	NULL	Description
HIDDEN_COLUMN	VARCHAR2(3)		Indicates whether the column is a hidden column (YES) or not (NO)
VIRTUAL_COLUMN	VARCHAR2(3)		Indicates whether the column is a virtual column (YES) or not (NO)
SEGMENT_COLUMN_ID	NUMBER		Sequence number of the column in the segment
INTERNAL_COLUMN_ID	NUMBER	NOT NULL	Internal sequence number of the column
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED
QUALIFIED_COL_NAME	VARCHAR2(4000)		Qualified column name

[1] These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views.

ALL_TAB_COLUMNS

ALL_TAB_COLUMNS describes the columns of the tables, views, and clusters accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement or the DBMS_STATS package.

Related Views

- DBA_TAB_COLUMNS describes the columns of all tables, views, and clusters in the database.
- USER_TAB_COLUMNS describes the columns of the tables, views, and clusters owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table, view, or cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table, view, or cluster
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Datatype of the column
DATA_TYPE_MOD	VARCHAR2(3)		Datatype modifier of the column
DATA_TYPE_OWNER	VARCHAR2(30)		Owner of the datatype of the column
DATA_LENGTH	NUMBER	NOT NULL	Length of the column (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatype; binary precision for FLOAT datatype, null for all other datatypes
DATA_SCALE	NUMBER		Digits to right of decimal point in a number
NULLABLE	VARCHAR2(1)		Specifies whether a column allows NULLs. Value is N if there is a NOT NULL constraint on the column or if the column is part of a PRIMARY KEY. The constraint should be in an ENABLE VALIDATE state.
COLUMN_ID	NUMBER		Sequence number of the column as created
DEFAULT_LENGTH	NUMBER		Length of default value for the column
DATA_DEFAULT	LONG		Default value for the column
NUM_DISTINCT	NUMBER		Number of distinct values in the column[1]
LOW_VALUE	RAW(32)		Low value in the column[1]
HIGH_VALUE	RAW(32)		High value in the column[1]

Column	Datatype	NULL	Description
DENSITY	NUMBER		Density of the column[1]
NUM_NULLS	NUMBER		Number of nulls in the column
NUM_BUCKETS	NUMBER		Number of buckets in the histogram for the column **Note:** The number of buckets in a histogram is specified in the SIZE parameter of the SQL statement ANALYZE. However, the Oracle Database does not create a histogram with more buckets than the number of rows in the sample. Also, if the sample contains any values that are very repetitious, the Oracle Database creates the specified number of buckets, but the value indicated by this column may be smaller because of an internal compression algorithm.
LAST_ANALYZED	DATE		Date on which this column was most recently analyzed
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this column
CHARACTER_SET_NAME	VARCHAR2(44)		Name of the character set: CHAR_CS or NCHAR_CS
CHAR_COL_DECL_LENGTH	NUMBER		Length
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether column statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
AVG_COL_LEN	NUMBER		Average length of the column (in bytes)
CHAR_LENGTH	NUMBER		Displays the length of the column in characters. This value only applies to the following datatypes: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR
CHAR_USED	VARCHAR2(1)		B | C. B indicates that the column uses BYTE length semantics. C indicates that the column uses CHAR length semantics. NULL indicates the datatype is not any of the following: ■ CHAR ■ VARCHAR2 ■ NCHAR ■ NVARCHAR2
V80_FMT_IMAGE	VARCHAR2(3)		Indicates whether the column data is in release 8.0 image format (YES) or not (NO)
DATA_UPGRADED	VARCHAR2(3)		Indicates whether the column data has been upgraded to the latest type version format (YES) or not (NO)
HISTOGRAM	VARCHAR2(15)		Indicates existence/type of histogram: ■ NONE ■ FREQUENCY ■ HEIGHT BALANCED

[1] These columns remain for backward compatibility with Oracle7. This information is now in the {TAB|PART}_COL_STATISTICS views.

ALL_TAB_COMMENTS

ALL_TAB_COMMENTS displays comments on the tables and views accessible to the current user.

Related Views

- DBA_TAB_COMMENTS displays comments on all tables and views in the database.
- USER_TAB_COMMENTS displays comments on the tables and views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
TABLE_TYPE	VARCHAR2(11)		Type of the object
COMMENTS	VARCHAR2(4000)		Comment on the object

See Also:

- "DBA_TAB_COMMENTS" on page 5-56
- "USER_TAB_COMMENTS" on page 5-132

ALL_TAB_HISTGRM_PENDING_STATS

ALL_TAB_HISTGRM_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions accessible to the current user.

Related Views

- DBA_TAB_HISTGRM_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions in the database.
- USER_TAB_HISTGRM_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Name of the owner
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
COLUMN_NAME	VARCHAR2(30)		Name of the column
ENDPOINT_NUMBER	NUMBER		Endpoint number
ENDPOINT_VALUE	NUMBER		Normalized endpoint value
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual endpoint value

ALL_TAB_HISTOGRAMS

ALL_TAB_HISTOGRAMS describes histograms on tables and views accessible to the current user.

Related Views

- DBA_TAB_HISTOGRAMS describes histograms on all tables and views in the database.
- USER_TAB_HISTOGRAMS describes histograms on all tables and views owned by the current user. This view does not display the OWNER column.

Note: These views are populated only if you collect statistics on the index using the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of table
TABLE_NAME	VARCHAR2(30)		Name of the table
COLUMN_NAME	VARCHAR2(4000)		Column name or attribute of the object type column
ENDPOINT_NUMBER	NUMBER		Histogram bucket number
ENDPOINT_VALUE	NUMBER		Normalized endpoint value for this bucket
ENDPOINT_ACTUAL_VALUE	VARCHAR2(1000)		Actual (not normalized) string value of the endpoint for this bucket

ALL_TAB_MODIFICATIONS

ALL_TAB_MODIFICATIONS describes tables accessible to the current user that have been modified since the last time statistics were gathered on the tables.

Related Views

- DBA_TAB_MODIFICATIONS provides such information for all tables in the database.
- USER_TAB_MODIFICATIONS provides such information for tables owned by the current user. This view does not display the TABLE_OWNER column.

Note: These views are populated only for tables with the MONITORING attribute. They are intended for statistics collection over a long period of time. For performance reasons, the Oracle Database does not populate these views immediately when the actual modifications occur. Run the FLUSH_DATABASE_MONITORING_INFO procedure in the DBMS_STATS PL/SQL package to populate these views with the latest information. The ANALYZE_ANY system privilege is required to run this procedure.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)		Owner of the modified table.
TABLE_NAME	VARCHAR2(30)		Name of the modified table
PARTITION_NAME	VARCHAR2(30)		Name of the modified partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the modified subpartition
INSERTS	NUMBER		Approximate number of inserts since the last time statistics were gathered
UPDATES	NUMBER		Approximate number of updates since the last time statistics were gathered
DELETES	NUMBER		Approximate number of deletes since the last time statistics were gathered
TIMESTAMP	DATE		Indicates the last time the table was modified
DROP_SEGMENTS	NUMBER		Number of partition and subpartition segments dropped since the last analyze

ALL_TAB_PARTITIONS

ALL_TAB_PARTITIONS provides partition-level partitioning information, partition storage parameters, and partition statistics collected by ANALYZE statements for partitions accessible to the current user.

Related Views

- DBA_TAB_PARTITIONS provides such information for all partitions in the database.
- USER_TAB_PARTITIONS provides such information for partitions of all partitioned objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COMPOSITE	VARCHAR2(3)		YES if the table is composite-partitioned, NO if it is not composite-partitioned
PARTITION_NAME	VARCHAR2(30)		Partition name
SUBPARTITION_COUNT	NUMBER		If this is a Local index on a table partitioned using a Composite method, the number of subpartitions in the partition
HIGH_VALUE	LONG		Partition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of partition bound value expression
PARTITION_POSITION	NUMBER	NOT NULL	Position of the partition within the table
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the partition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes (for a range partition); size of the initial extent in blocks (for a composite partition)
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes (for a range partition); size of secondary extents in blocks (for a composite partition)
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size
FREELISTS	NUMBER		Number of process freelists allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of partition
COMPRESSION	VARCHAR2(8)		Indicates whether this partition is compressed (ENABLED) or not (DISABLED)
NUM_ROWS	NUMBER		Number of rows in the partition
BLOCKS	NUMBER		Number of used blocks in the partition
EMPTY_BLOCKS	NUMBER		Number of empty (never used) blocks in the partition
AVG_SPACE	NUMBER		Average available free space in the partition
CHAIN_CNT	NUMBER		Number of chained rows in the partition

Column	Datatype	NULL	Description
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this partition
LAST_ANALYZED	DATE		Date on which this partition was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool to be used for the partition blocks
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were collected for the partition as a whole (YES) or were estimated from statistics on underlying subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

ALL_TAB_PENDING_STATS

ALL_TAB_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions accessible to the current user.

Related Views

- DBA_TAB_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions in the database.
- USER_TAB_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Name of the owner
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
NUM_ROWS	NUMBER		Number of rows
BLOCKS	NUMBER		Number of blocks
AVG_ROW_LEN	NUMBER		Average row length
SAMPLE_SIZE	NUMBER		Sample size
LAST_ANALYZED	DATE		Time of last analyze operation

ALL_TAB_PRIVS

ALL_TAB_PRIVS describes the following types of grants:

- Object grants for which the current user is the object owner, grantor, or grantee
- Object grants for which an enabled role or PUBLIC is the grantee

Related Views

- DBA_TAB_PRIVS describes all object grants in the database.
- USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also:

- "DBA_TAB_PRIVS" on page 5-57
- "USER_TAB_PRIVS" on page 5-133

ALL_TAB_PRIVS_MADE

ALL_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner or grantor.

Related View

USER_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner. This view does not display the OWNER column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: "USER_TAB_PRIVS_MADE" on page 5-133

ALL_TAB_PRIVS_RECD

ALL_TAB_PRIVS_RECD describes the following types of grants:

- Object grants for which the current user is the grantee
- Object grants for which an enabled role or PUBLIC is the grantee

Related View

USER_TAB_PRIVS_RECD describes the object grants for which the current user is the grantee. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: "USER_TAB_PRIVS_RECD" on page 5-133

ALL_TAB_STAT_PREFS

ALL_TAB_STAT_PREFS provides information about statistics preferences for tables accessible to the current user.

Related Views

- DBA_TAB_STAT_PREFS provides information about statistics preferences for tables in the database.
- USER_TAB_STAT_PREFS provides information about statistics preferences for tables owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
PREFERENCE_NAME	VARCHAR2(30)	NOT NULL	Name of the preference
PREFERENCE_VALUE	VARCHAR2(1000)		Value of the preference

ALL_TAB_STATISTICS

ALL_TAB_STATISTICS displays optimizer statistics for the tables accessible to the current user.

Related Views

- DBA_TAB_STATISTICS displays optimizer statistics for all tables in the database.
- USER_TAB_STATISTICS displays optimizer statistics for the tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition
PARTITION_POSITION	NUMBER		Position of the partition within the table
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition within the partition

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(12)		Type of the object: - TABLE - PARTITION - SUBPARTITION
NUM_ROWS	NUMBER		Number of rows in the object
BLOCKS	NUMBER		Number of used blocks in the object
EMPTY_BLOCKS	NUMBER		Number of empty blocks in the object
AVG_SPACE	NUMBER		Average available free space in the object
CHAIN_CNT	NUMBER		Number of chained rows in the object
AVG_ROW_LEN	NUMBER		Average row length, including row overhead
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
AVG_CACHED_BLOCKS	NUMBER		Average number of blocks in the buffer cache
AVG_CACHE_HIT_RATIO	NUMBER		Average cache hit ratio for the object
SAMPLE_SIZE	NUMBER		Sample size used in analyzing the table
LAST_ANALYZED	DATE		Date of the most recent time the table was analyzed
GLOBAL_STATS	VARCHAR2(3)		Indicates whether statistics were calculated without merging underlying partitions (YES) or not (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
STATTYPE_LOCKED	VARCHAR2(5)		Type of statistics lock: - DATA - CACHE - ALL
STALE_STATS	VARCHAR2(3)		Indicates whether statistics for the object are stale or not.

See Also:

- "DBA_TAB_STATISTICS" on page 5-57
- "USER_TAB_STATISTICS" on page 5-133

ALL_TAB_STATS_HISTORY

ALL_TAB_STATS_HISTORY provides a history of table statistics modifications for all tables accessible to the current user.

Related Views

- DBA_TAB_STATS_HISTORY provides a history of table statistics modifications for all tables in the database.
- USER_TAB_STATS_HISTORY provides a history of table statistics modifications for all tables owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
TABLE_NAME	VARCHAR2(30)		Name of the table
PARTITION_NAME	VARCHAR2(30)		Name of the partition

Column	Datatype	NULL	Description
SUBPARTITION_NAME	VARCHAR2(30)		Name of the subpartition
STATS_UPDATE_TIME	TIMESTAMP(6) WITH TIME ZONE		Time at which the statistics were updated

See Also:

- "DBA_TAB_STATS_HISTORY" on page 5-58
- "USER_TAB_STATS_HISTORY" on page 5-133

ALL_TAB_SUBPARTITIONS

ALL_TAB_SUBPARTITIONS describes, for each table subpartition accessible to the current user, the subpartition name, name of the table and partition to which it belongs, and its storage attributes.

Related Views

- DBA_TAB_SUBPARTITIONS provides such information for all subpartitions in the database.
- USER_TAB_SUBPARTITIONS provides such information for subpartitions of all partitioned objects owned by the current user. This view does not display the TABLE_OWNER column.

Note: Statistics are not collected on a per-subpartition basis.

Column	Datatype	NULL	Description
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
PARTITION_NAME	VARCHAR2(30)		Partition name
HIGH_VALUE	LONG		Subpartition bound value expression
HIGH_VALUE_LENGTH	NUMBER	NOT NULL	Length of subpartition bound value expression
SUBPARTITION_NAME	VARCHAR2(30)		Subpartition name
SUBPARTITION_POSITION	NUMBER	NOT NULL	Position of a subpartition within a partition
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the subpartition
PCT_FREE	NUMBER	NOT NULL	Minimum percentage of free space in a block
PCT_USED	NUMBER	NOT NULL	Minimum percentage of used space in a block
INI_TRANS	NUMBER	NOT NULL	Initial number of transactions
MAX_TRANS	NUMBER	NOT NULL	Maximum number of transactions
INITIAL_EXTENT	NUMBER		Size of the initial extent in bytes (for a range partition); size of the initial extent in blocks (for a composite partition)
NEXT_EXTENT	NUMBER		Size of secondary extents in bytes (for a range partition); size of secondary extents in blocks (for a composite partition)
MIN_EXTENT	NUMBER	NOT NULL	Minimum number of extents allowed in the segment
MAX_EXTENT	NUMBER	NOT NULL	Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER	NOT NULL	Percentage increase in extent size

Column	Datatype	NULL	Description
FREELISTS	NUMBER		Number of freelist groups allocated in this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated in this segment
LOGGING	VARCHAR2(3)		Logging attribute of subpartition
COMPRESSION	VARCHAR2(8)		Indicates whether this subpartition is compressed (ENABLED) or not (DISABLED)
NUM_ROWS	NUMBER		The number of rows
BLOCKS	NUMBER		The number of blocks
EMPTY_BLOCKS	NUMBER		The number of empty blocks
AVG_SPACE	NUMBER		The average space
CHAIN_CNT	NUMBER		The chain count
AVG_ROW_LEN	NUMBER		The average row length
SAMPLE_SIZE	NUMBER		The sample size
LAST_ANALYZED	DATE		The date on which this table was most recently analyzed
BUFFER_POOL	VARCHAR2(7)		The default buffer pool for this subpartition
GLOBAL_STATS	VARCHAR2(3)		Indicates whether column statistics for the subpartition were collected by analyzing the table as a whole (YES) or estimated from statistics collected for partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)

ALL_TABLES

ALL_TABLES describes the relational tables accessible to the current user. To gather statistics for this view, use the ANALYZE SQL statement.

Related Views

- DBA_TABLES describes all relational tables in the database.
- USER_TABLES describes the relational tables owned by the current user. This view does not display the OWNER column.

Note: Columns marked with an asterisk (*) are populated only if you collect statistics on the table with the ANALYZE statement or the DBMS_STATS package.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the table; null for partitioned, temporary, and index-organized tables
CLUSTER_NAME	VARCHAR2(30)		Name of the cluster, if any, to which the table belongs
IOT_NAME	VARCHAR2(30)		Name of the index-organized table, if any, to which the overflow or mapping table entry belongs. If the IOT_TYPE column is not null, then this column contains the base table name.
STATUS	VARCHAR2(8)		If a previous DROP TABLE operation failed, indicates whether the table is unusable (UNUSABLE) or valid (VALID)

Column	Datatype	NULL	Description
PCT_FREE	NUMBER		Minimum percentage of free space in a block; null for partitioned tables
PCT_USED	NUMBER		Minimum percentage of used space in a block; null for partitioned tables
INI_TRANS	NUMBER		Initial number of transactions; null for partitioned tables
MAX_TRANS	NUMBER		Maximum number of transactions; null for partitioned tables
INITIAL_EXTENT	NUMBER		Size of the initial extent (in bytes); null for partitioned tables
NEXT_EXTENT	NUMBER		Size of secondary extents (in bytes); null for partitioned tables
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment; null for partitioned tables
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment; null for partitioned tables
PCT_INCREASE	NUMBER		Percentage increase in extent size; null for partitioned tables
FREELISTS	NUMBER		Number of process freelists allocated to the segment; null for partitioned tables
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to the segment; null for partitioned tables
LOGGING	VARCHAR2(3)		Logging attribute; NULL for partitioned tables
BACKED_UP	VARCHAR2(1)		Has table been backed up since last change
NUM_ROWS*	NUMBER		Number of rows in the table
BLOCKS*	NUMBER		Number of used data blocks in the table
EMPTY_BLOCKS	NUMBER		Number of empty (never used) data blocks in the table. This column is populated only if you collect statistics on the table using the ANALYZE statement.
AVG_SPACE*	NUMBER		Average amount of free space, in bytes, in a data block allocated to the table
CHAIN_CNT*	NUMBER		Number of rows in the table that are chained from one data block to another, or which have migrated to a new block, requiring a link to preserve the old ROWID
AVG_ROW_LEN*	NUMBER		Average length of a row in the table (in bytes)
AVG_SPACE_FREELIST_BLOCKS	NUMBER		Average freespace of all blocks on a freelist
NUM_FREELIST_BLOCKS	NUMBER		Number of blocks on the freelist
DEGREE	VARCHAR2(10)		Number of threads per instance for scanning the table
INSTANCES	VARCHAR2(10)		Number of instances across which the table is to be scanned
CACHE	VARCHAR2(5)		Indicates whether the table is to be cached in the buffer cache (Y) or not (N)
TABLE_LOCK	VARCHAR2(8)		Indicates whether table locking is enabled (ENABLED) or disabled (DISABLED)
SAMPLE_SIZE	NUMBER		Sample size used in analyzing this table
LAST_ANALYZED	DATE		Date on which this table was most recently analyzed
PARTITIONED	VARCHAR2(3)		Indicates whether this table is partitioned. Set to YES if it is partitioned.
IOT_TYPE	VARCHAR2(12)		If this is an index-organized table, then IOT_TYPE is IOT, IOT_OVERFLOW, or IOT_MAPPING. If this is not an index-organized table, then IOT_TYPE is NULL.

Column	Datatype	NULL	Description
TEMPORARY	VARCHAR2(1)		Can the current session only see data that it place in this object itself?
SECONDARY	VARCHAR2(1)		Whether the trigger is a secondary object created by the ODCIIndexCreate method of the Oracle Data Cartridge (Y |N)
NESTED	VARCHAR2(3)		Indicates whether the table is a nested table (YES) or not (NO)
BUFFER_POOL	VARCHAR2(7)		The default buffer pool for the object. NULL for partitioned tables
ROW_MOVEMENT	VARCHAR2(8)		Whether partitioned row movement is enabled or disabled
GLOBAL_STATS	VARCHAR2(3)		For partitioned tables, indicates whether statistics were collected for the table as a whole (YES) or were estimated from statistics on underlying partitions and subpartitions (NO)
USER_STATS	VARCHAR2(3)		Indicates whether statistics were entered directly by the user (YES) or not (NO)
DURATION	VARCHAR2(15)		Indicates the duration of a temporary table: ■ SYS$SESSION: the rows are preserved for the duration of the session ■ SYS$TRANSACTION: the rows are deleted after COMMIT Null for a permanent table
SKIP_CORRUPT	VARCHAR2(8)		Whether the Oracle Database ignores blocks marked corrupt during table and index scans (ENABLED) or raises an error (DISABLED). To enable this feature, run the DBMS_REPAIR.SKIP_CORRUPT_BLOCKS procedure.
MONITORING	VARCHAR2(3)		Whether the table has the MONITORING attribute set
CLUSTER_OWNER	VARCHAR2(30)		Owner of the cluster, if any, to which the table belongs
DEPENDENCIES	VARCHAR2(8)		Indicates whether row-level dependency tracking is enabled (ENABLED) or disabled (DISABLED)
COMPRESSION	VARCHAR2(8)		Indicates whether table compression is enabled (ENABLED) or not (DISABLED); null for partitioned tables
DROPPED	VARCHAR2(3)		Indicates whether the table has been dropped and is in the recycle bin (YES) or not (NO); null for partitioned tables

See Also:

- "DBA_TABLES" on page 5-58
- "USER_TABLES" on page 5-134

ALL_TRIGGER_COLS

ALL_TRIGGER_COLS describes the use of columns in the triggers accessible to the current user and in triggers on tables accessible to the current user. If the user has the CREATE ANY TRIGGER privilege, then this view describes the use of columns in all triggers in the database.

Related Views

- DBA_TRIGGER_COLS describes the use of columns in all triggers in the database.

- USER_TRIGGER_COLS describes the use of columns in the triggers owned by the current user and in triggers on tables owned by the current user.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)		Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)		Name of the trigger
TABLE_OWNER	VARCHAR2(30)		Owner of the table on which the trigger is defined
TABLE_NAME	VARCHAR2(30)		Table on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the column used in the trigger
COLUMN_LIST	VARCHAR2(3)		Indicates whether the column is specified in the UPDATE clause (YES) or not (NO)
COLUMN_USAGE	VARCHAR2(17)		How the column is used in the trigger: ▪ NEW IN ▪ OLD IN ▪ NEW IN OLD IN ▪ NEW OUT ▪ NEW IN OUT ▪ NEW OUT OLD IN ▪ NEW IN OUT OLD IN ▪ PARENT IN

See Also:

- "DBA_TRIGGER_COLS" on page 5-62
- "USER_TRIGGER_COLS" on page 5-134

ALL_TRIGGER_ORDERING

ALL_TRIGGER_ORDERING describes the triggers accessible to the current user that have FOLLOWS or PRECEDES ordering.

Related Views

- DBA_TRIGGER_ORDERING describes all triggers in the database that have FOLLOWS or PRECEDES ordering.
- USER_TRIGGER_ORDERING describes the triggers owned by the current user that have FOLLOWS or PRECEDES ordering. This view does not display the TRIGGER_OWNER column.

Column	Datatype	NULL	Description
TRIGGER_OWNER	VARCHAR2(30)	NOT NULL	Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)		Name of the trigger
REFERENCED_TRIGGER_OWNER	VARCHAR2(30)		Owner of the referenced trigger
REFERENCED_TRIGGER_NAME	VARCHAR2(30)		Name of the referenced trigger
ORDERING_TYPE	VARCHAR2(8)		Type of the ordering between the trigger and the referenced trigger: ▪ FOLLOWS ▪ PRECEDES

See Also:

- "DBA_TRIGGER_ORDERING" on page 5-62
- "USER_TRIGGER_ORDERING" on page 5-134

ALL_TRIGGERS

ALL_TRIGGERS describes the triggers on tables accessible to the current user. If the user has the CREATE ANY TRIGGER privilege, then this view describes all triggers in the database.

Related Views

- DBA_TRIGGERS describes all triggers in the database.
- USER_TRIGGERS describes the triggers owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the trigger
TRIGGER_NAME	VARCHAR2(30)		Name of the trigger
TRIGGER_TYPE	VARCHAR2(16)		When the trigger fires: ■ BEFORE STATEMENT ■ BEFORE EACH ROW ■ AFTER STATEMENT ■ AFTER EACH ROW ■ INSTEAD OF ■ COMPOUND
TRIGGERING_EVENT	VARCHAR2(227)		DML, DDL, or database event that fires the trigger **See Also:** The CREATE TRIGGER statement in *Oracle Database SQL Language Reference* for a listing of triggering events
TABLE_OWNER	VARCHAR2(30)		Owner of the table on which the trigger is defined
BASE_OBJECT_TYPE	VARCHAR2(16)		Base object on which the trigger is defined: ■ TABLE ■ VIEW ■ SCHEMA ■ DATABASE
TABLE_NAME	VARCHAR2(30)		If the base object type of the trigger is SCHEMA or DATABASE, then this column is NULL; if the base object type of the trigger is TABLE or VIEW, then this column indicates the table or view name on which the trigger is defined
COLUMN_NAME	VARCHAR2(4000)		Name of the nested table column (if a nested table trigger), else NULL
REFERENCING_NAMES	VARCHAR2(128)		Names used for referencing OLD and NEW column values from within the trigger
WHEN_CLAUSE	VARCHAR2(4000)		Must evaluate to TRUE for TRIGGER_BODY to execute
STATUS	VARCHAR2(8)		Indicates whether the trigger is enabled (ENABLED) or disabled (DISABLED)
DESCRIPTION	VARCHAR2(4000)		Trigger description; useful for re-creating a trigger creation statement

Column	Datatype	NULL	Description
ACTION_TYPE	VARCHAR2(11)		Action type of the trigger body: - CALL - PL/SQL
TRIGGER_BODY	LONG		Statements executed by the trigger when it fires

See Also:

- "DBA_TRIGGERS" on page 5-62
- "USER_TRIGGERS" on page 5-134

ALL_TYPE_ATTRS

ALL_TYPE_ATTRS describes the attributes of the object types accessible to the current user.

Related Views

- DBA_TYPE_ATTRS describes the attributes of all object types in the database.
- USER_TYPE_ATTRS describes the attributes of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)		Name of the type
ATTR_NAME	VARCHAR2(30)		Name of the attribute
ATTR_TYPE_MOD	VARCHAR2(7)		Type modifier of the attribute: - REF - POINTER
ATTR_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the attribute
ATTR_TYPE_NAME	VARCHAR2(30)		Name of the type of the attribute
LENGTH	NUMBER		Length of the CHAR attribute, or maximum length of the VARCHAR or VARCHAR2 attribute.
PRECISION	NUMBER		Decimal precision of the NUMBER or DECIMAL attribute, or binary precision of the FLOAT attribute.
SCALE	NUMBER		Scale of the NUMBER or DECIMAL attribute
CHARACTER_SET _NAME	VARCHAR2(44)		Character set name of the attribute (CHAR_CS or NCHAR_CS)
ATTR_NO	NUMBER		Syntactical order number or position of the attribute as specified in the type specification or CREATE TYPE statement (not to be used as an ID number)
INHERITED	VARCHAR2(3)		Indicates whether the attribute is inherited from a supertype (YES) or not (NO)

See Also:

- "DBA_TYPE_ATTRS" on page 5-64
- "USER_TYPE_ATTRS" on page 5-135

ALL_TYPE_METHODS

ALL_TYPE_METHODS describes the methods of the object types accessible to the current user.

Related Views

- DBA_TYPE_METHODS describes the methods of all object types in the database.
- USER_TYPE_METHODS describes the methods of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
METHOD_NAME	VARCHAR2(30)	NOT NULL	Name of the method
METHOD_NO	NUMBER	NOT NULL	Method number for distinguishing overloaded methods (not to be used as ID number)
METHOD_TYPE	VARCHAR2(6)		Type of the method: ■ MAP ■ ORDER ■ PUBLIC
PARAMETERS	NUMBER	NOT NULL	Number of parameters to the method
RESULTS	NUMBER	NOT NULL	Number of results returned by the method
FINAL	VARCHAR2(3)		Indicates whether the method is final (YES) or not (NO)
INSTANTIABLE	VARCHAR2(3)		Indicates whether the method is instantiable (YES) or not (NO)
OVERRIDING	VARCHAR2(3)		Indicates whether the method is overriding a supertype method (YES) or not (NO)
INHERITED	VARCHAR2(3)		Indicates whether the method is inherited from a supertype (YES) or not (NO)

See Also:

- "DBA_TYPE_METHODS" on page 5-64
- "USER_TYPE_METHODS" on page 5-135

ALL_TYPE_VERSIONS

ALL_TYPE_VERSIONS describes the versions of the object types accessible to the current user.

Related Views

- DBA_TYPE_VERSIONS describes the versions of all object types in the database.
- USER_TYPE_VERSIONS describes the versions of the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the type
TYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the type
VERSION#	NUMBER	NOT NULL	Internal version number of the type

Column	Datatype	NULL	Description
TYPECODE	VARCHAR2(30)		Typecode of the type
STATUS	VARCHAR2(7)		Status of the type: ■ N/A ■ VALID ■ INVALID
LINE	NUMBER	NOT NULL	Line number of the type's spec
TEXT	VARCHAR2(4000)		Text of the type's spec
HASHCODE	RAW(17)		Hashcode of the type

See Also:

- "DBA_TYPE_VERSIONS" on page 5-64
- "USER_TYPE_VERSIONS" on page 5-135

ALL_TYPES

ALL_TYPES describes the object types accessible to the current user.

Related Views

- DBA_TYPES describes all object types in the database.
- USER_TYPES describes the object types owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the type
TYPE_NAME	VARCHAR2(30)		Name of the type
TYPE_OID	RAW(16)		Object identifier (OID) of the type
TYPECODE	VARCHAR2(30)		Typecode of the type
ATTRIBUTES	NUMBER		Number of attributes (if any) in the type
METHODS	NUMBER		Number of methods (if any) in the type
PREDEFINED	VARCHAR2(3)		Indicates whether the type is a predefined type (YES) or not (NO)
INCOMPLETE	VARCHAR2(3)		Indicates whether the type is an incomplete type (YES) or not (NO)
FINAL	VARCHAR2(3)		Indicates whether the type is a final type (YES) or not (NO)
INSTANTIABLE	VARCHAR2(3)		Indicates whether the type is an instantiable type (YES) or not (NO)
SUPERTYPE_OWNER	VARCHAR2(30)		Owner of the supertype (null if type is not a subtype)
SUPERTYPE_NAME	VARCHAR2(30)		Name of the supertype (null if type is not a subtype)
LOCAL_ATTRIBUTES	NUMBER		Number of local (not inherited) attributes (if any) in the subtype
LOCAL_METHODS	NUMBER		Number of local (not inherited) methods (if any) in the subtype
TYPEID	RAW(16)		Type ID value of the type

See Also:

- "DBA_TYPES" on page 5-64
- "USER_TYPES" on page 5-135

ALL_UNUSED_COL_TABS

ALL_UNUSED_COL_TABS describes the tables accessible to the current user that contain unused columns.

Related Views

- DBA_UNUSED_COL_TABS describes all tables in the database that contain unused columns.
- USER_UNUSED_COL_TABS describes the tables owned by the current user that contain unused columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
COUNT	NUMBER		Number of unused columns

See Also:

- "DBA_UNUSED_COL_TABS" on page 5-65
- "USER_UNUSED_COL_TABS" on page 5-135

ALL_UPDATABLE_COLUMNS

ALL_UPDATABLE_COLUMNS describes all columns in a join view that are updatable by the current user, subject to appropriate privileges.

Related Views

- DBA_UPDATABLE_COLUMNS describes all columns in a join view that are updatable by the database administrator, subject to appropriate privileges.
- USER_UPDATABLE_COLUMNS describes all columns owned by the current user that are in a join view and are updatable by the current user, subject to appropriate privileges.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Table owner
TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
UPDATABLE	VARCHAR2(3)		Indicates whether the column is updatable
INSERTABLE	VARCHAR2(3)		Indicates whether the column is insertable
DELETABLE	VARCHAR2(3)		Indicates whether the column is deletable

ALL_USERS

ALL_USERS lists all users of the database visible to the current user. This view does not describe the users (see the related views).

Related Views

- DBA_USERS describes all users of the database, and contains more columns than ALL_USERS.
- USER_USERS describes the current user, and contains more columns than ALL_USERS.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
CREATED	DATE	NOT NULL	User creation date

See Also:

- "DBA_USERS" on page 5-65
- "USER_USERS" on page 5-136

ALL_USTATS

ALL_USTATS describes the user-defined statistics collected on the tables and indexes accessible to the current user.

Related Views

- DBA_USTATS describes the user-defined statistics collected on all tables and indexes in the database.
- USER_USTATS describes the user-defined statistics collected on the tables and indexes owned by the current user.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or index for which the statistics have been collected
OBJECT_NAME	VARCHAR2(30)		Name of the table or index for which the statistics have been collected
PARTITION_NAME	VARCHAR2(30)		Partition name of a table; NULL if the table is either non-partitioned or the entry corresponds to the aggregate statistics for the table
OBJECT_TYPE	VARCHAR2(6)		Type of the object for which statistics have been collected: ■ INDEX ■ COLUMN
ASSOCIATION	VARCHAR2(8)		Statistics type association: ■ DIRECT Direct association with the object for which the statistics have been collected ■ IMPLICIT - Association for which the statistics have been collected is with the column type or index type, and the object is an instance of that column type or index type

Column	Datatype	NULL	Description
COLUMN_NAME	VARCHAR2(30)		Column name, if OBJECT_TYPE is COLUMN, for which statistics have been collected
STATSTYPE_SCHEMA	VARCHAR2(30)		Schema of the statistics type which was used to collect the statistics
STATSTYPE_NAME	VARCHAR2(30)		Name of the statistics type which was used to collect statistics
STATISTICS	RAW(2000)		User-collected statistics for the object

See Also:

- "DBA_USTATS" on page 5-66
- "USER_USTATS" on page 5-136

ALL_VARRAYS

ALL_VARRAYS describes the varrays accessible to the current user.

Related Views

- DBA_VARRAYS describes all varrays in the database.
- USER_VARRAYS describes the varrays owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the table containing the varray
PARENT_TABLE_NAME	VARCHAR2(30)		Name of the containing table
PARENT_TABLE_COLUMN	VARCHAR2(4000)		Name of the varray column or attribute
TYPE_OWNER	VARCHAR2(30)		Owner of the varray type
TYPE_NAME	VARCHAR2(30)		Name of the varray type
LOB_NAME	VARCHAR2(30)		Name of the LOB if the varray is stored in a LOB
STORAGE_SPEC	VARCHAR2(30)		Indicates whether the storage was defaulted (DEFAULT) or user-specified (USER_SPECIFIED)
RETURN_TYPE	VARCHAR2(20)		Return type of the column: ■ LOCATOR ■ VALUE
ELEMENT_SUBSTITUTABLE	VARCHAR2(25)		Indicates whether the varray element is substitutable (Y) or not (N)

See Also:

- "DBA_VARRAYS" on page 5-66
- "USER_VARRAYS" on page 5-136

ALL_VIEWS

ALL_VIEWS describes the views accessible to the current user.

Related Views

- DBA_VIEWS describes all views in the database.

- USER_VIEWS describes the views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the view
VIEW_NAME	VARCHAR2(30)	NOT NULL	Name of the view
TEXT_LENGTH	NUMBER		Length of the view text
TEXT	LONG		View text
TYPE_TEXT_LENGTH	NUMBER		Length of the type clause of the typed view
TYPE_TEXT	VARCHAR2(4000)		Type clause of the typed view
OID_TEXT_LENGTH	NUMBER		Length of the WITH OID clause of the typed view
OID_TEXT	VARCHAR2(4000)		WITH OID clause of the typed view
VIEW_TYPE_OWNER	VARCHAR2(30)		Owner of the type of the view if the view is a typed view
VIEW_TYPE	VARCHAR2(30)		Type of the view if the view is a typed view
SUPERVIEW_NAME	VARCHAR2(30)		Name of the superview
EDITIONING_VIEW	VARCHAR2(1)		Reserved for future use
READ_ONLY	VARCHAR2(1)		Indicates whether the view is read-only (Y) or not (N)

See Also:

- "DBA_VIEWS" on page 5-67
- "USER_VIEWS" on page 5-136

ALL_WARNING_SETTINGS

ALL_WARNING_SETTINGS displays information about the warning parameter settings for the objects accessible to the current user.

Related Views

- DBA_WARNING_SETTINGS displays information about the warning parameter settings for all objects in the database.
- USER_WARNING_SETTINGS displays information about the warning parameter settings for the objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
OBJECT_ID	NUMBER	NOT NULL	Object number of the object
OBJECT_TYPE	VARCHAR2(12)		Type of the object: ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY

Column	Datatype	NULL	Description
WARNING	VARCHAR2(40)		Warning number or category: - INFORMATIONAL - PERFORMANCE - SEVERE - ALL
SETTING	VARCHAR2(7)		Value of the warning setting: - DISABLE - ENABLE - ERROR

See Also:

- "DBA_WARNING_SETTINGS" on page 5-67
- "USER_WARNING_SETTINGS" on page 5-136

ALL_XML_INDEXES

ALL_XML_INDEXES describes the XML indexes accessible to the current user.

Related Views

- DBA_XML_INDEXES describes all XML indexes in the database.
- USER_XML_INDEXES describes the XML indexes owned by the current user. This view does not display the INDEX_OWNER column.

Column	Datatype	NULL	Description
INDEX_OWNER	VARCHAR2(30)	NOT NULL	Owner of the XML index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the XML index
TABLE_OWNER	VARCHAR2(30)	NOT NULL	Owner of the indexed object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the indexed object
PATH_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the path table
PARAMETERS	XMLTYPE		Indexed paths and Scheduler job information
ASYNC	VARCHAR2(9)		Asynchronous index type: - ON-COMMIT - MANUAL - EVERY - ALWAYS
STALE	VARCHAR2(5)		Indicates whether the index type is stale (TRUE) or not (FALSE)
PEND_TABLE_NAME	VARCHAR2(30)		Name of the pending table
TYPE	VARCHAR2(10)		Type of the indexed column: - REPOSITORY - CSX - CLOB in OR - CLOB

Column	Datatype	NULL	Description
EX_OR_INCLUDE	VARCHAR2(8)		Path subsetting: - INCLUDE - EXCLUDE - FULLY IX

See Also:

- "DBA_XML_INDEXES" on page 5-73
- "USER_XML_INDEXES" on page 5-136

ALL_XML_SCHEMAS

ALL_XML_SCHEMAS describes the registered XML schemas accessible to the current user.

Related Views

- DBA_XML_SCHEMAS describes all registered XML schemas in the database.
- USER_XML_SCHEMAS describes the registered XML schemas owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML schema
SCHEMA_URL	VARCHAR2(700)		Schema URL of the XML schema
LOCAL	VARCHAR2(3)		Indicates whether the XML schema is local (YES) or global (NO)
SCHEMA	XMLTYPE		XML schema document
INT_OBJNAME	VARCHAR2(4000)		Internal database object name for the schema
QUAL_SCHEMA_URL	VARCHAR2(767)		Fully qualified schema URL
HIER_TYPE	VARCHAR2(11)		Type of hierarchy for which the schema is enabled: - NONE - RESMETADATA - CONTENTS
BINARY	VARCHAR2(3)		Indicates whether the XML Schema is registered for binary encoding usage (YES) or not (NO)
SCHEMA_ID	RAW(16)		Opaque schema identifier (16 bytes)
HIDDEN	VARCHAR2(3)		Indicates whether the XML Schema has been deleted in hidden mode (YES) or not (NO)

See Also:

- "DBA_XML_SCHEMAS" on page 5-73
- "USER_XML_SCHEMAS" on page 5-136

ALL_XML_TAB_COLS

ALL_XML_TAB_COLS describes the columns of the XML tables accessible to the current user.

Related Views

- DBA_XML_TAB_COLS describes the columns of all XML tables in the database.
- USER_XML_TAB_COLS describes the columns of the XML tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML table
TABLE_NAME	VARCHAR2(30)		Name of the XML table
COLUMN_NAME	VARCHAR2(4000)		Name of the XML table column
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2(2000)		Name of the XML SChema element that is used for the table
STORAGE_TYPE	VARCHAR2(17)		Storage option for the XMLtype data: ■ OBJECT-RELATIONAL ■ BINARY ■ CLOB
ANYSCHEMA	VARCHAR2(3)		If storage is BINARY, indicates whether the column allows ANYSCHEMA (YES) or not (NO), else NULL
NONSCHEMA	VARCHAR2(3)		If storage is BINARY, indicates whether the column allows NONSCHEMA (YES) or not (NO), else NULL

See Also:

- "DBA_XML_TAB_COLS" on page 5-73
- "USER_XML_TAB_COLS" on page 5-136

ALL_XML_TABLES

ALL_XML_TABLES describes the XML tables accessible to the current user.

Related Views

- DBA_XML_TABLES describes all XML tables in the database.
- USER_XML_TABLES describes the XML tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML table
TABLE_NAME	VARCHAR2(30)		Name of the XML table
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2(2000)		Name of the XML SChema element that is used for the table

Column	Datatype	NULL	Description
STORAGE_TYPE	VARCHAR2(17)		Storage option for the XMLtype data: ■ OBJECT-RELATIONAL ■ BINARY ■ CLOB
ANYSCHEMA	VARCHAR2(3)		If storage is BINARY, indicates whether the column allows ANYSCHEMA (YES) or not (NO), else NULL
NONSCHEMA	VARCHAR2(3)		If storage is BINARY, indicates whether the column allows NONSCHEMA (YES) or not (NO), else NULL

See Also:

- "DBA_XML_TABLES" on page 5-73
- "USER_XML_TABLES" on page 5-137

ALL_XML_VIEW_COLS

ALL_XML_VIEW_COLS describes the columns of the XML views accessible to the current user.

Related Views

- DBA_XML_VIEW_COLS describes the columns of all XML views in the database.
- USER_XML_VIEW_COLS describes the columns of the XML views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML view
VIEW_NAME	VARCHAR2(30)		Name of the XML view
COLUMN_NAME	VARCHAR2(4000)		Name of the XML view column
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2(2000)		Name of the XML SChema element that is used for the view

See Also:

- "DBA_XML_VIEW_COLS" on page 5-73
- "USER_XML_VIEW_COLS" on page 5-137

ALL_XML_VIEWS

ALL_XML_VIEWS describes the XML views accessible to the current user.

Related Views

- DBA_XML_VIEWS describes all XML views the database.
- USER_XML_VIEWS describes the XML views owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML view
VIEW_NAME	VARCHAR2(30)		Name of the XML view
XMLSCHEMA	VARCHAR2(700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2(30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2(2000)		Name of the XML SChema element that is used for the view

See Also:

- "DBA_XML_VIEWS" on page 5-73
- "USER_XML_VIEWS" on page 5-137

AUDIT_ACTIONS

AUDIT_ACTIONS describes audit trail action type codes. This table can be used to map action type numbers to action type names.

Column	Datatype	NULL	Description
ACTION	NUMBER	NOT NULL	Numeric audit trail action type code. These values also appear in the COMMAND column of the V$SESSION dynamic performance view, and are listed with that view. **See Also:** Table 8–2 on page 8-12 for a listing of the action type codes
NAME	VARCHAR2(28)	NOT NULL	Name of the type of audit trail action

CAT

CAT is a synonym for USER_CATALOG.

See Also: "USER_CATALOG" on page 5-102

CATALOG

CATALOG is included for compatibility. Oracle recommends that you not use this view.

CHAINED_ROWS

CHAINED_ROWS stores the output for the ANALYZE statement with the LIST CHAINED ROWS clause. You must run the utlchain.sql or utlchn1.sql script to create this table.

Column	Description
OWNER_NAME	Table owner
TABLE_NAME	Table name
CLUSTER_NAME	Cluster the table is in, if any
PARTITION_NAME	The name of the partition
SUBPARTITION_NAME	The name of the subpartition

Column	Description
HEAD_ROWID	ROWID the chained row is accessed by
ANALYZE_TIMESTAMP	Date/time that the ANALYZE statement was issued

CHANGE_PROPAGATION_SETS

CHANGE_PROPAGATION_SETS is a synonym for ALL_CHANGE_PROPAGATION_SETS.

See Also: "ALL_CHANGE_PROPAGATION_SETS" on page 2-24

CHANGE_PROPAGATIONS

CHANGE_PROPAGATIONS is a synonym for ALL_CHANGE_PROPAGATIONS.

See Also: "ALL_CHANGE_PROPAGATIONS" on page 2-25

CHANGE_SETS

CHANGE_SETS is a synonym for ALL_CHANGE_SETS.

See Also: "ALL_CHANGE_SETS" on page 2-25

CHANGE_SOURCES

CHANGE_SOURCES is a synonym for ALL_CHANGE_SOURCES.

See Also: "ALL_CHANGE_SOURCES" on page 2-26

CHANGE_TABLES

CHANGE_TABLES is a synonym for ALL_CHANGE_TABLES.

See Also: "ALL_CHANGE_TABLES" on page 2-26

CLIENT_RESULT_CACHE_STATS$

CLIENT_RESULT_CACHE_STATS$ displays various Client Result Cache settings and usage statistics.

Column	Datatype	NULL	Description
CACHE_ID	NUMBER	NOT NULL	Unique ID per client cache
STAT_ID	NUMBER	NOT NULL	Statistic ID
NAME	VARCHAR2(128)		Name of the statistic (see Table 3–1)
VALUE	NUMBER		Value of the statistic

Table 3–1 CLIENT_RESULT_CACHE_STATS$ Statistics

Statistic Name	Description
Block Size	Size (in bytes) of each memory block in the result cache.
Block Count Max	Maximum number of blocks that can be allocated in the result cache based on the cache size configuration parameters on server and on client.
Block Count Current	Current number of blocks allocated by the client result cache.

Table 3–1 (Cont.) CLIENT_RESULT_CACHE_STATS$ Statistics

Statistic Name	Description
Hash Bucket Count	Size of the hash table used for query matching.
Create Count Success	Number of cached result sets that did not get invalidated prior to caching all the rows of the result set.
Create Count Failure	Number of cached result sets that did not fetch all the rows in the result set.
Find Count	Number of cache hits.
Invalidation Count	Number of cached result sets that got invalidated due to database changes that could have affected the result set.
Delete Count Invalid	Number of cached result rests not invalidated whose memory was reclaimed by result cache.
Delete Count Valid	Number of invalidated cached result rests whose memory was reclaimed by result cache.

CLU

CLU is a synonym for USER_CLUSTERS.

See Also: "USER_CLUSTERS" on page 5-103

COL

COL is included for compatibility. Oracle recommends that you not use this view.

COLS

COLS is a synonym for USER_TAB_COLUMNS.

See Also: "USER_TAB_COLUMNS" on page 5-131

DATABASE_PROPERTIES

DATABASE_PROPERTIES lists Permanent database properties.

Column	Datatype	NULL	Description
PROPERTY_NAME	VARCHAR2(30)	NOT NULL	Property name
PROPERTY_VALUE	VARCHAR2(4000)		Property value
DESCRIPTION	VARCHAR2(4000)		Property description

4

Static Data Dictionary Views: DBA_2PC_ NEIGHBORS to DBA_MVIEWS

This chapter contains the static data dictionary views DBA_2PC_NEIGHBORS through DBA_MVIEWS.

DBA_2PC_NEIGHBORS

DBA_2PC_NEIGHBORS describes incoming and outgoing connections for pending transactions.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)		Local identifier of a transaction
IN_OUT	VARCHAR2(3)		IN for incoming connections, OUT for outgoing
DATABASE	VARCHAR2(128)		IN for client database name, OUT for outgoing database link
DBUSER_OWNER	VARCHAR2(30)		IN for name of local user, OUT for owner of database link
INTERFACE	VARCHAR2(1)		C for request commit, otherwise N for prepare or request readonly commit
DBID	VARCHAR2(16)		Database ID at the other end of the connection
SESS#	NUMBER(38)		Session number of the connection at this database
BRANCH	VARCHAR2(128)		Transaction branch ID of the connection at this database

DBA_2PC_PENDING

DBA_2PC_PENDING describes distributed transactions awaiting recovery.

Column	Datatype	NULL	Description
LOCAL_TRAN_ID	VARCHAR2(22)	NOT NULL	String of form: n.n.n; n is a number
GLOBAL_TRAN_ID	VARCHAR2(169)		Globally unique transaction ID
STATE	VARCHAR2(16)	NOT NULL	Collecting, prepared, committed, forced commit, or forced rollback
MIXED	VARCHAR2(3)		YES indicates part of the transaction committed and part rolled back
ADVICE	VARCHAR2(1)		C for commit, R for rollback, else NULL
TRAN_COMMENT	VARCHAR2(255)		Text for commit work comment text
FAIL_TIME	DATE	NOT NULL	Value of SYSDATE when the row was inserted (transaction or system recovery)
FORCE_TIME	DATE		Time of manual force decision (null if not forced locally)
RETRY_TIME	DATE	NOT NULL	Time automatic recovery (RECO) last tried to recover the transaction
OS_USER	VARCHAR2(64)		Operating system-specific name for the end-user
OS_TERMINAL	VARCHAR2(255)		Operating system-specific name for the end-user terminal
HOST	VARCHAR2(128)		Name of the host machine for the end-user
DB_USER	VARCHAR2(30)		Oracle user name of the end-user at the topmost database
COMMIT#	VARCHAR2(16)		Global commit number for committed transactions

DBA_ADDM_FDG_BREAKDOWN

DBA_ADDM_FDG_BREAKDOWN describes the contribution for each finding from the different instances. There is one row for each finding and for each instance participating in the analysis. Rows are omitted if the impact from that instance is not sufficient to register a finding in a local ADDM analysis.

Related View

USER_ADDM_FDG_BREAKDOWN describes the contribution for each finding from the different instances owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER	NOT NULL	Identifies the task to which this finding belongs (see DBA_ADVISOR_TASKS)
FINDING_ID	NUMBER	NOT NULL	Identifies the finding (see DBA_ADVISOR_FINDINGS)
INSTANCE_NUMBER	NUMBER	NOT NULL	The number of the instance contributing to the finding
DATABASE_TIME	NUMBER		The database time, in microseconds, accumulated by this instance during the analysis period
ACTIVE_SESSIONS	NUMBER		The average number of active sessions of the finding in this instance
PERC_ACTIVE_SESSIONS	NUMBER		The percentage of contribution from this instance compared to the total impact of the finding

DBA_ADDM_FINDINGS

DBA_ADDM_FINDINGS displays the ADDM findings discovered by all advisors in the database. Each row for ADDM tasks in the related DBA_ADVISOR_FINDINGS view has a corresponding row in this view.

Related View

USER_ADDM_FINDINGS displays the ADDM findings discovered by the advisors owned by the current user. Each row for ADDM tasks in the related USER_ADVISOR_FINDINGS view has a corresponding row in this view. The USER_ADDM_FINDINGS view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
FINDING_ID	NUMBER	NOT NULL	Identifier of the finding
FINDING_NAME	VARCHAR2(4000)		Name of the finding
TYPE	VARCHAR2(11)		Type of the finding: PROBLEM SYMPTOM ERROR INFORMATION
PARENT	NUMBER	NOT NULL	Identifier of the parent finding
OBJECT_ID	NUMBER		Identifier of the associated object, if any
IMPACT_TYPE	VARCHAR2(4000)		Impact of the finding on the system
IMPACT	NUMBER		Impact value
MESSAGE	VARCHAR2(4000)		Message describing the finding
MORE_INFO	VARCHAR2(4000)		Additional info associated with the finding
FILTERED	CHAR(1)		A value of Y means that the row in the view was filtered out by a directive (or a combination of directives). A value of N means that the row was not filtered.
DATABASE_TIME	NUMBER		The database time, in microseconds, accumulated by this finding
ACTIVE_SESSIONS	NUMBER		The average number of active sessions for the finding
PERC_ACTIVE_SESS	NUMBER		The percentage of active sessions for this finding out of the total active sessions for the task
IS_AGGREGATE	CHAR(1)		A value of Y means that this finding is created for global/continental ADDM as an aggregate of local ADDM findings. Otherwise, the value is N.
METER_LEVEL	VARCHAR2(6)		???

DBA_ADDM_INSTANCES

DBA_ADDM_INSTANCES provides instance-level information for ADDM tasks that finished executing. For each instance that was supposed to be analyzed (whether it was or not) there is one row describing information about it.

Related View

USER_ADDM_INSTANCES provides instance-level information for ADDM tasks that finished executing in all instances owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER	NOT NULL	The ID of the main ADDM task

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER	NOT NULL	The number of the instance
INSTANCE_NAME	VARCHAR2(16)		The name of the instance
HOST_NAME	VARCHAR2(64)		The name of the machine on which the instance was running
STATUS	VARCHAR2(10)		How information from this instance was used by the ADDM task. A value of ANALYZED means that the instance participated fully in the analysis. For the following remaining values, the instance was not used during task execution, for the stated reason: BOUNCED - the instance was shut down or started during the analysis period NO_SNAPS - there were either begin or end snapshots missing for the instance NO_STATS - there were key statistics missing for the instance NOT_FOUND - no mention of this instance could be found in AWR during the analysis period
DATABASE_TIME	NUMBER		The database time, in microseconds, accumulated by this instance during the analysis period
ACTIVE_SESSIONS	NUMBER		The average number of active sessions for the instance during the analysis period
PERC_ACTIVE_SESS	NUMBER		The percentage of active sessions for this instance, out of the total active sessions for the task
METER_LEVEL	VARCHAR2(6)		???
LOCAL_TASK_ID	NUMBER		The ID of a local ADDM task that contained an analysis of the instance for the same analysis period as that of the main task. If the main task is a local ADDM, then this value is the same as the TASK_ID value.

DBA_ADDM_TASKS

DBA_ADDM_TASKS displays information about all ADDM tasks in the database. The view contains one row for each row in the related DBA_ADVISOR_TASKS view that has ADVISOR_NAME=ADDM and STATUS=COMPLETED.

Related View

USER_ADDM_TASKS displays information about the ADDM tasks owned by the current user. The view contains one row for each row in the related USER_ADVISOR_TASKS view that has ADVISOR_NAME=ADDM and STATUS=COMPLETED. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
CREATED	DATE	NOT NULL	Creation date of the task
LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
PARENT_TASK_ID	NUMBER		Identifier of the parent task (if the task was created as a result of the recommendation of another task)

Column	Datatype	NULL	Description
PARENT_RXEC_ID	NUMBER		Identifier of the recommendation within the parent task that resulted in the creation of the task
LAST_EXECUTION	VARCHAR2(30)		Name of the current or last task execution
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.
EXECUTION_TYPE#	NUMBER		???
EXECUTION_DESCRIPTION	VARCHAR2(256)		Optional description of the last execution
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_END	DATE		Execution end date and time of the task
STATUS	VARCHAR2(11)		Current operational status of the task: ■ INITIAL - Initial state of the task; no recommendations are present ■ EXECUTING - Task is currently running ■ INTERRUPTED - Task analysis was interrupted by the user. Recommendation data, if present, can be viewed and reported at this time. ■ COMPLETED - Task successfully completed the analysis operation. Recommendation data can be viewed and reported. ■ ERROR - An error occurred during the analysis operation. Recommendations, if present, can be viewed and reported at this time.
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor, regarding the status
PCT_COMPLETION_TIME	NUMBER		Percent completion, in terms of time, of the task when it is executing
PROGRESS_METRIC	NUMBER		Metric that measures the progress of the task in terms of quality. Each advisor may have its own metric.
METRIC_UNITS	VARCHAR2(64)		Unit of the metric used to measure progress
ACTIVITY_COUNTER	NUMBER		Counter that is updated frequently by the advisor, denoting that useful work is being performed
RECOMMENDATION_COUNT	NUMBER		Number of recommendations produced
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition
SOURCE	VARCHAR2(30)		Optional name that identifies the creator of the task
HOW_CREATED	VARCHAR2(30)		Optional task or template on which the object was based
READ_ONLY	VARCHAR2(5)		Indicates whether the task is read-only (TRUE) or not (FALSE)
SYSTEM_TASK	VARCHAR2(5)		???
ADVISOR_ID	NUMBER	NOT NULL	Unique identifier for the advisor
STATUS#	NUMBER		???
DBID	NUMBER		The database ID that the task was using
DBNAME	VARCHAR2(9)		The name of the database that the task was analyzing
DIVERSION	VARCHAR2(17)		???
ANALYSIS_VERSION	VARCHAR2(17)		The version of the database that executed the task
BEGIN_SNAP_ID	NUMBER		The snapshot ID that starts the analysis period
BEGIN_TIME	TIMESTAMP(3)		The SYSDATE at the time the BEGIN_SNAP_ID was taken
END_SNAP_ID	NUMBER		The snapshot ID that ends the analysis period
END_TIME	TIMESTAMP(3)		The SYSDATE at the time the END_SNAP_ID was taken

Column	Datatype	NULL	Description
REQUESTED_ANALYSIS	VARCHAR2(8)		The type of ADDM analysis that was requested before task execution, as follows: DATABASE - global ADDM INSTANCE - local ADDM PARTIAL - continental ADDM
ACTUAL_ANALYSIS	VARCHAR2(8)		The type of ADDM analysis that was actually performed when the task was executed (either DATABASE, INSTANCE, or PARTIAL)
DATABASE_TIME	NUMBER		The total database time accumulated in the analysis period (and analyzed instances) in microseconds
ACTIVE_SESSIONS	NUMBER		The average active sessions during the analysis period (and analyzed sessions)
METER_LEVEL	VARCHAR2(6)		???

DBA_ADVISOR_ACTIONS

DBA_ADVISOR_ACTIONS displays information about the actions associated with all recommendations in the database. Each action is specified by the COMMAND and ATTR1 through ATTR6 columns. Each command defines how the attribute columns will be used.

Related View

USER_ADVISOR_ACTIONS displays information about the actions associated with the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
REC_ID	NUMBER	NOT NULL	Recommendation associated with the action
ACTION_ID	NUMBER	NOT NULL	Unique identifier for the action
OBJECT_ID	NUMBER		Object associated with the action
COMMAND	VARCHAR2(64)		Command to be executed **See Also:** DBA_ADVISOR_COMMANDS for a list of commands
COMMAND_ID	NUMBER	NOT NULL	ID of the command to be executed **See Also:** DBA_ADVISOR_COMMANDS for a list of commands
FLAGS	NUMBER		Advisor-specific flags
ATTR1	VARCHAR2(4000)		Parameters defining the command
ATTR2	VARCHAR2(4000)		Parameters defining the command
ATTR3	VARCHAR2(4000)		Parameters defining the command
ATTR4	VARCHAR2(4000)		Parameters defining the command
ATTR5	CLOB		Parameters defining the command; to be used if the text is significantly large (for example, a SQL statement defining a materialized view)

Column	Datatype	NULL	Description
ATTR6	CLOB		Parameters defining the command; to be used if the text is significantly large (for example, a SQL statement defining a materialized view)
NUM_ATTR1	NUMBER		General numeric attribute
NUM_ATTR2	NUMBER		General numeric attribute
NUM_ATTR3	NUMBER		General numeric attribute
NUM_ATTR4	NUMBER		General numeric attribute
NUM_ATTR5	NUMBER		General numeric attribute
MESSAGE	VARCHAR2(4000)		Message associated with the action
FILTERED	VARCHAR2(1)		A value of Y means that the row in the view was filtered out by a directive (or a combination of directives). A value of N means that the row was not filtered.

See Also: "USER_ADVISOR_ACTIONS" on page 5-97

DBA_ADVISOR_COMMANDS

DBA_ADVISOR_COMMANDS displays information about the commands used by all advisors in the database for specifying recommendation actions. In addition to the set of commands in the COMMAND column of V$SESSION, the following additional commands are defined:

- RUN ADVISOR
- CHECK EXECUTION PLAN
- ALTER PARAMETER
- ENABLE TRACE

Column	Datatype	NULL	Description
COMMAND_ID	NUMBER		Identifier of the command
COMMAND_NAME	VARCHAR2(64)		Name of the command

DBA_ADVISOR_DEF_PARAMETERS

DBA_ADVISOR_DEF_PARAMETERS displays all default task parameters and their current values in the database. When a task or object is created, the parameters and their values are copied into the private parameter table.

Column	Datatype	NULL	Description
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	Name of the advisor that supports the parameter
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	Value of the parameter. Numeric parameter values are converted to a string equivalent. Possible keywords as values: ■ ALL ■ UNLIMITED ■ UNUSED

Column	Datatype	NULL	Description
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter: ■ NUMBER - Numeric value ■ STRING - String value. If the string contains special characters, then it will be enclosed in single quotes. ■ STRINGLIST - Comma-separated list of string elements. If a string element contains a comma or other special characters, then the element will be enclosed in single quotes. ■ TABLE - Single table reference. A reference will contain a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported. ■ TABLELIST - List of one or more comma-separated table references. A reference will contain schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported.
IS_DEFAULT	VARCHAR2(1)		Indicates whether the parameter value is set to the advisor's default value (Y) or not (N)
IS_OUTPUT	VARCHAR2(1)		Indicates whether the task execution process sets the parameter value (Y) or not (N)
IS_MODIFIABLE_ANYTIME	VARCHAR2(1)		Indicates whether the parameter value can be modified when the task is not in its initial state (Y) or not (N)
IS_SYSTEM_TASK_ONLY	VARCHAR2(1)		???
DESCRIPTION	VARCHAR2(4000)		Optional description of the parameter
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.

DBA_ADVISOR_DEFINITIONS

DBA_ADVISOR_DEFINITIONS displays the properties of all advisors in the database. The view contains one row for each task, representing the current state of the task as well as execution-specific data such as progress monitoring and completion status.

Column	Datatype	NULL	Description
ADVISOR_ID	NUMBER	NOT NULL	Unique identifier for the advisor
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	Name of the advisor
PROPERTY	NUMBER	NOT NULL	Properties: ■ Bit 0: - Indicates whether the advisor runs in COMPREHENSIVE mode (1) or not (0) ■ Bit 1: - Indicates whether the advisor runs in LIMITED mode (1) or not (0) ■ Bit 2: - Indicates whether the advisor is resumable (1) or not (0) ■ Bit 3: - Indicates whether the advisor accepts user directives (1) or not (0)

DBA_ADVISOR_DIR_DEFINITIONS

DBA_ADVISOR_DIR_DEFINITIONS provides a definition of the base directive.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Unique id for directive. The directive management engine automatically generates ID numbers. The identifier is unique among all directives regardless of the domain name and directive name.
ADVISOR_ID	NUMBER	NOT NULL	Identifier number of the owner advisor.
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	The name of the advisor to which this directive belongs.
DIRECTIVE_NAME	VARCHAR2(30)	NOT NULL	Any value that further classifies this directive within a domain. The domain and the name form a unique key for the directive.
DOMAIN_NAME	VARCHAR2(30)	NOT NULL	Domain or namespace name
DESCRIPTION	VARCHAR2(256)	NOT NULL	An optional description that documents the purpose of the directive.
TYPE	NUMBER	NOT NULL	Further describes the use of the directive. Possible values are: 1. **Filter** - An Xpath filter 2. **Single Value** - Evaluation returns a single string value 3. **Multiple Values** - Evaluation returns one to many string values 4. **Conditional** - Evaluation returns a single value based on an input key, similar to a CASE or SWITCH statement
TYPE_NAME	VARCHAR2(15)		A decoded version of the TYPE column.
TASK_STATUS	VARCHAR2(9)		The status of the directive instances when a task is not in its initial state. Possible values are: - IMMUTABLE - MUTABLE
INSTANCES	VARCHAR2(8)		Indicates whether a directive will permit multiple instances. Possible values are: - SINGLE - MULTIPLE
METADATA	CLOB	NOT NULL	A DTD that is used to process the directive.A DTD that is used to process the directive.

DBA_ADVISOR_DIR_INSTANCES

DBA_ADVISOR_DIR_INSTANCES provides information about all global instances for a directive.

Column	Datatype	NULL	Description
DIRECTIVE_ID	NUMBER	NOT NULL	Unique id for directive. The directive management engine automatically generates ID numbers.
INSTANCE_ID	NUMBER	NOT NULL	Unique id for the directive instance. The directive management engine automatically generates ID numbers.
INSTANCE_NAME	VARCHAR2(30)	NOT NULL	A user-assigned name for the directive instance.
DATA	CLOB	NOT NULL	An XML document that gives meaningful default values for all parts of the directive.

DBA_ADVISOR_DIR_TASK_INST

DBA_ADVISOR_DIR_TASK_INST provides information about all task directive instances.

Related View

USER_ADVISOR_DIR_TASK_INST provides information about all task directive instances owned by the current user.

Column	Datatype	NULL	Description
DIRECTIVE_ID	NUMBER	NOT NULL	Unique id for directive. The directive management engine automatically generates ID numbers.
SEQ_ID	NUMBER	NOT NULL	Unique id for the directive instance. The directive management engine automatically generates ID numbers.
INSTANCE_NAME	VARCHAR2(30)	NOT NULL	A user-assigned name for the directive instance.
USERNAME	VARCHAR2(30)	NOT NULL	Database user who owns the task instance.
TASK_ID	NUMBER		An advisor task identifier to which the directive instance is associated
TASK_NAME	VARCHAR2(30)		An advisor task to which the directive instance is associated.
DATA	CLOB	NOT NULL	An XML document that gives meaningful default values for all parts of the directive.

DBA_ADVISOR_EXECUTIONS

DBA_ADVISOR_EXECUTIONS displays metadata information for task executions. For example, the SQL Replay Advisor creates a minimum of three executions to perform a change impact analysis on a SQL workload. The first one collects performance data for the version of the workload before the change, the second one collects data for the version of the workload after the change, and the third one performs impact analysis. All of these executions belong to the same task and are grouped into this view.

Related View

USER_ADVISOR_EXECUTIONS displays metadata information for task executions owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	The name of the task execution with which this entry (row) is associated
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.
EXECUTION_TYPE#	NUMBER		???
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
EXECUTION_END	DATE		Execution end date and time of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
ADVISOR_ID	NUMBER	NOT NULL	Unique identifier for the advisor
STATUS	VARCHAR2(11)		Current operational status of the task
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor, regarding the status

Column	Datatype	NULL	Description
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition

DBA_ADVISOR_EXECUTION_TYPES

DBA_ADVISOR_EXECUTION_TYPES displays possible execution action for a given advisor. Only advisors that support multiple executions of their tasks have entries in this view.

Column	Datatype	NULL	Description
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	Name of the advisor
EXECUTION_TYPE	VARCHAR2(30)	NOT NULL	Type of the last execution. This information is optional for single-execution tasks.
EXECUTION_DESCRIPTION	VARCHAR2(4000)		Optional description of the last execution.

DBA_ADVISOR_EXEC_PARAMETERS

DBA_ADVISOR_EXEC_PARAMETERS displays task parameters that are specific to a given task execution. Different executions of a task can have different parameter settings..

Related View

USER_ADVISOR_EXEC_PARAMETERS displays task parameters that are specific to a given task execution owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	The name of the task execution with which this entry (row) is associated
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAMETER_VALUE	VARCHAR2(4000)		Value of the parameter. Numeric parameter values are converted to a string equivalent.
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter
IS_DEFAULT	VARCHAR2(1)		Indicates whether the parameter value is set to the advisor's default value (Y) or not (N)
IS_OUTPUT	VARCHAR2(1)		Indicates whether the task execution process sets the parameter value (Y) or not (N)
IS_MODIFIABLE_ANYTIME	VARCHAR2(1)		Indicates whether the parameter value can be modified when the task is not in its initial state (Y) or not (N)
DESCRIPTION	VARCHAR2(4000)		Optional description of the parameter
PARAMETER_FLAGS	NUMBER	NOT NULL	???
PARAMETER_TYPE#	NUMBER	NOT NULL	???

DBA_ADVISOR_FDG_BREAKDOWN

DBA_ADVISOR_FDG_BREAKDOWN ???.

Related View

USER_ADVISOR_FDG_BREAKDOWN ???.

Column	Datatype	NULL	Description
TASK_ID	NUMBER	NOT NULL	???
FINDING_ID	NUMBER	NOT NULL	???
INSTANCE_NUMBER	NUMBER	NOT NULL	???
IMPACT	NUMBER		???
PERC_IMPACT	NUMBER		???
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated

DBA_ADVISOR_FINDING_NAMES

DBA_ADVISOR_FINDING_NAMES provides a list of all finding names registered with the Advisor Framework.

Column	Datatype	NULL	Description
ID	NUMBER		ID of the finding name
ADVISOR_NAME	VARCHAR2(30)	NOT NULL	Advisor name
FINDING_NAME	VARCHAR2(4000)		Finding name

DBA_ADVISOR_FINDINGS

DBA_ADVISOR_FINDINGS displays the findings discovered by all advisors in the database.

Related View

USER_ADVISOR_FINDINGS displays the findings discovered by the advisors owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
FINDING_ID	NUMBER	NOT NULL	Identifier of the finding
FINDING_NAME	VARCHAR2(4000)		Name of the finding
TYPE	VARCHAR2(11)		Type of the finding: ▪ PROBLEM ▪ SYMPTOM ▪ ERROR ▪ INFORMATION
PARENT	NUMBER	NOT NULL	Identifier of the parent finding
OBJECT_ID	NUMBER		Identifier of the associated object, if any
IMPACT_TYPE	VARCHAR2(4000)		Impact of the finding on the system
IMPACT	NUMBER		Impact value

Column	Datatype	NULL	Description
MESSAGE	VARCHAR2(4000)		Message describing the finding
MORE_INFO	VARCHAR2(4000)		Additional info associated with the finding
FILTERED	VARCHAR2(1)		A value of Y means that the row in the view was filtered out by a directive (or a combination of directives). A value of N means that the row was not filtered.

See Also: "USER_ADVISOR_FINDINGS" on page 5-97

DBA_ADVISOR_JOURNAL

DBA_ADVISOR_JOURNAL displays the journal entries for all tasks in the database.

Related View

USER_ADVISOR_JOURNAL displays the journal entries for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task or workload object
TASK_NAME	VARCHAR2(30)		Name of the task or workload object
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
JOURNAL_ENTRY_SEQ	NUMBER	NOT NULL	Sequence number of the journal entry (unique for each task). This sequence number is used to order the data.
JOURNAL_ENTRY_TYPE	VARCHAR2(12)		Type of the task: ■ FATAL ■ ERROR ■ WARNING ■ INFORMATION ■ INFORMATION[2 | 3 | 4 | 5 | 6]
JOURNAL_ENTRY	VARCHAR2(4000)		Entry in the journal

See Also: "USER_ADVISOR_JOURNAL" on page 5-98

DBA_ADVISOR_LOG

DBA_ADVISOR_LOG displays information about the current state of all tasks in the database, as well as execution-specific data such as progress monitoring and completion status. The view contains one row for each task.

Related View

USER_ADVISOR_LOG displays information about the current state of the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task

Column	Datatype	NULL	Description
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_END	DATE		Execution end date and time of the task
STATUS	VARCHAR2(11)		Current operational status of the task: ■ INITIAL - Initial state of the task; no recommendations are present ■ EXECUTING - Task is currently running ■ COMPLETED - Task successfully completed the analysis operation. Recommendation data can be viewed and reported. ■ INTERRUPTED - Task analysis was interrupted by the user. Recommendation data, if present, can be viewed and reported at this time. ■ CANCELLED ■ FATAL ERROR - A fatal error occurred during the analysis operation. All recommendation data is unusable.
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor regarding the status
PCT_COMPLETION_TIME	NUMBER		Percent completion, in terms of time, of the task when it is executing
PROGRESS_METRIC	NUMBER		Metric that measures the progress of the task in terms of quality. Each advisor could have its own metric.
METRIC_UNITS	VARCHAR2(64)		Unit of the metric used to measure progress
ACTIVITY_COUNTER	NUMBER		Counter that is updated frequently by the advisor, denoting that useful work is being performed
RECOMMENDATION_COUNT	NUMBER		Number of recommendations produced
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition

See Also: "USER_ADVISOR_LOG" on page 5-98

DBA_ADVISOR_OBJECT_TYPES

DBA_ADVISOR_OBJECT_TYPES displays information about the object types used by all advisors in the database. In addition to the regular database object types (such as TABLE and INDEX), the following types are defined:

- SYSTEM
- I/O
- SGA
- PGA
- SHARED POOL
- BUFFER CACHE
- LIBRARY CACHE
- PROCESS
- SESSION
- ENQUEUE
- LATCH

- ROLLBACK SEGMENT
- FILE
- PARAMETER
- CURSOR
- SQL
- SQL WORKLOAD

Column	Datatype	NULL	Description
OBJECT_TYPE_ID	NUMBER		Type identifier
OBJECT_TYPE	VARCHAR2(64)		Type name

DBA_ADVISOR_OBJECTS

DBA_ADVISOR_OBJECTS displays information about the objects currently referenced by all advisors in the database. Each row in the view pertains to an object instantiation.

Related View

USER_ADVISOR_OBJECTS displays information about the objects currently referenced by the advisors owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
OBJECT_ID	NUMBER	NOT NULL	Identifier of the object
TYPE	VARCHAR2(64)		Name of the type
TYPE_ID	NUMBER	NOT NULL	Type identifier number
TASK_ID	NUMBER	NOT NULL	Task referencing the object
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
ATTR1	VARCHAR2(4000)		Attributes and identifier of the object
ATTR2	VARCHAR2(4000)		Attributes and identifier of the object
ATTR3	VARCHAR2(4000)		Attributes and identifier of the object
ATTR4	CLOB		Attributes and identifiers that cannot be expressed in the ATTR1, ATTR2, and ATTR3 columns
ATTR5	VARCHAR2(4000)		Attributes and identifier of the object
ATTR6	RAW(2000)		Attributes and identifier of the object
ATTR7	NUMBER		Attributes and identifier of the object
ATTR8	NUMBER		Attributes and identifier of the object
ATTR9	NUMBER		Attributes and identifier of the object
ATTR10	NUMBER		Attributes and identifier of the object
OTHER	CLOB		Other attributes and identifiers of the object

Note: The definition of the ATTRn columns depends on the advisors that are using the object. For example, the SQL object type defines the attribute columns as follows:

- ATTR1 contains the SQL hash value
- ATTR2 contains the SQL address (in the cursor cache)
- ATTR4 contains the SQL text

See Also: "USER_ADVISOR_OBJECTS" on page 5-98

DBA_ADVISOR_PARAMETERS

DBA_ADVISOR_PARAMETERS displays all task parameters and their current values in the database. This data is accessible by all tasks.

Related View

USER_ADVISOR_PARAMETERS displays the task parameters and their current values for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task or workload object
TASK_ID	NUMBER	NOT NULL	Unique identifier number of the task or workload object
TASK_NAME	VARCHAR2(30)		Name of the task or workload object
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	Value of the parameter. Numeric parameter values are converted to a string equivalent. Possible keywords as values: - ALL - UNLIMITED - UNUSED
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter: - NUMBER - Numeric value - STRING - String value. If the string contains special characters, then it will be enclosed in single quotes. - STRINGLIST - Comma-separated list of string elements. If a string element contains a comma or other special characters, then the element will be enclosed in single quotes. - TABLE - Single table reference. A reference will contain a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported. - TABLELIST - List of one or more comma-separated table references. A reference will contain schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported.
IS_DEFAULT	VARCHAR2(1)		Indicates whether the parameter value is set to the advisor's default value (Y) or not (N)

Column	Datatype	NULL	Description
IS_OUTPUT	VARCHAR2(1)		Indicates whether the task execution process sets the parameter value (Y) or not (N)
IS_MODIFIABLE_ANYTIME	VARCHAR2(1)		Indicates whether the parameter value can be modified when the task is not in its initial state (Y) or not (N)
DESCRIPTION	VARCHAR2(4000)		Optional description of the parameter
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.

See Also: "USER_ADVISOR_PARAMETERS" on page 5-98

DBA_ADVISOR_PARAMETERS_PROJ

DBA_ADVISOR_PARAMETERS_PROJ ???.

Column	Datatype	NULL	Description
TASK_ID	NUMBER	NOT NULL	???
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	???
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	???
PARAMETER_TYPE	VARCHAR2(10)		???
IS_DEFAULT	VARCHAR2(1)		???
IS_OUTPUT	VARCHAR2(1)		???
IS_MODIFIABLE_ANYTIME	VARCHAR2(1)		???
IS_SYSTEM_TASK_ONLY	VARCHAR2(1)		???
DESCRIPTION	VARCHAR2(4000)		???

DBA_ADVISOR_RATIONALE

DBA_ADVISOR_RATIONALE displays information about the rationales for all recommendations in the database.

Related View

USER_ADVISOR_RATIONALE displays information about the rationales for the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
REC_ID	NUMBER		Recommendation associated with the rationale
RATIONALE_ID	NUMBER	NOT NULL	Unique identifier for the rationale
IMPACT_TYPE	VARCHAR2(4000)		Impact on the system due to the problem described in the rationale. The impact can be described in terms of time, cost, or % degradation.
IMPACT	NUMBER		Calculated impact value
MESSAGE	VARCHAR2(4000)		Message containing an overview of the rationale

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER		Identifier of an object specified in the DBA_ADVISOR_OBJECTS view
TYPE	VARCHAR2(30)		Type of the rationale; defines what data exists in the attribute columns and how to interpret it: ■ TEXT - Text sentence for descriptive messages. The ATTR1 column contains the text. ■ CHART - Chart containing data to be displayed. The ATTR1 column contains the data.
ATTR1	VARCHAR2(4000)		Parameters defining the rationale
ATTR2	VARCHAR2(4000)		Parameters defining the rationale
ATTR3	VARCHAR2(4000)		Parameters defining the rationale
ATTR4	VARCHAR2(4000)		Parameters defining the rationale
ATTR5	CLOB		Parameters defining the rationale

See Also: "USER_ADVISOR_RATIONALE" on page 5-98

DBA_ADVISOR_RECOMMENDATIONS

DBA_ADVISOR_RECOMMENDATIONS displays the results of an analysis of all recommendations in the database. A recommendation can have multiple actions associated with it. Actions are described in the DBA_ADVISOR_ACTIONS view. A recommendation also points to a set of rationales that present a justification/reasoning for that recommendation. These rationales are in the DBA_ADVISOR_RATIONALE view.

Related View

USER_ADVISOR_RECOMMENDATIONS displays the results of an analysis of the recommendations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
REC_ID	NUMBER	NOT NULL	Unique identifier of the recommendation
TASK_ID	NUMBER	NOT NULL	Task that owns the recommendation
TASK_NAME	VARCHAR2(30)		Name of the task
EXECUTION_NAME	VARCHAR2(30)		The name of the task execution with which this entry (row) is associated
FINDING_ID	NUMBER		Unique identifier of the finding
TYPE	VARCHAR2(30)		Type of the recommendation
RANK	NUMBER		Ranking, in terms of importance, within the set of recommendations generated for the task
PARENT_REC_IDS	VARCHAR2(4000)		Comma-separated list of the recommendation IDs of the parent recommendations. If this column is nonzero, then the recommendation depends on the parents, and cannot be accepted if the parents are not accepted.

Column	Datatype	NULL	Description
BENEFIT_TYPE	VARCHAR2(4000)		Describes the benefit obtained by carrying out the recommendation If there is a set of parents for the recommendation, then the benefit is the cumulative benefit (the improvement in system performance when this and all prior parent recommendations are accepted). If there are no parents, then this is the improvement when the recommendation is accepted, independent of other recommendations.
BENEFIT	NUMBER		Calculated benefit value
ANNOTATION_STATUS	VARCHAR2(11)		When a task is complete, the recommendations are marked ACCEPT. The status can be changed later using the MARK_RECOMMENDATION procedure: ■ ACCEPT - Current recommendation is ready to implement. This recommendation can also be used as advice for future analysis operations. ■ REJECT - Current recommendation is not acceptable to the user, and therefore will be excluded from any implementation scripts. This recommendation can also be used as advice for future analysis operations. ■ IGNORE - Though not rejected, the current recommendation will be ignored when generating scripts and will never be used as advice to future analysis operations.
FLAGS	NUMBER		Advisor-specific flags
FILTERED	VARCHAR2(1)		A value of Y means that the row in the view was filtered out by a directive (or a combination of directives). A value of N means that the row was not filtered.

See Also: "USER_ADVISOR_RECOMMENDATIONS" on page 5-98

DBA_ADVISOR_SQLA_REC_SUM

DBA_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for all workload objects in the database after an Access Advisor analysis operation.

Related View

USER_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER		Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
REC_ID	NUMBER		Identifier of the recommendation
TOTAL_STMTS	NUMBER		Total number of statements processed during analysis
TOTAL_PRECOST	NUMBER		Total cost of executing the statements in which the recommended object will be utilized, prior to the recommendations
TOTAL_POSTCOST	NUMBER		Total cost of executing the statements in which the recommended object will be utilized, after the recommendations have been implemented

See Also: "USER_ADVISOR_SQLA_REC_SUM" on page 5-98

DBA_ADVISOR_SQLA_TABLES

DBA_ADVISOR_SQLA_TABLES ???.

Related View

USER_ADVISOR_SQLA_TABLES ???.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		???
TASK_ID	NUMBER		???
TASK_NAME	VARCHAR2(30)		???
SQL_ID	VARCHAR2(13)		???
STMT_ID	NUMBER		???
TABLE_OWNER	VARCHAR2(30)		???
TABLE_NAME	VARCHAR2(30)		???

DBA_ADVISOR_SQLA_WK_MAP

DBA_ADVISOR_SQLA_WK_MAP displays the workload references for all tasks in the database. Workload references are necessary to allow the SQL Access Advisor to find required workload data.

Related View

USER_ADVISOR_SQLA_WK_MAP displays the workload references for the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER		Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
WORKLOAD_ID	NUMBER		Unique identifier of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
IS_STS	NUMBER		???

See Also: "USER_ADVISOR_SQLA_WK_MAP" on page 5-99

DBA_ADVISOR_SQLA_WK_STMTS

DBA_ADVISOR_SQLA_WK_STMTS displays information about all workload objects in the database after an Access Advisor analysis operation.

Related View

USER_ADVISOR_SQLA_WK_STMTS displays information about the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_NAME	VARCHAR2(30)		Name of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
SQLSET_ID	NUMBER		???
SQLSET_NAME	VARCHAR2(30)		???
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
SQL_ID	NUMBER	NOT NULL	Generated identifier of the statement
SQL_SEQ	NUMBER	NOT NULL	???
PLAN_HASH_VALUE	NUMBER	NOT NULL	???
PARSING_SCHEMA_NAME	VARCHAR2(30)		???
USERNAME	VARCHAR2(30)		Name of the user executing the statement
MODULE	VARCHAR2(64)		Name of the module issuing the statement
ACTION	VARCHAR2(64)		Module action for the statement
CPU_TIME	NUMBER		Total CPU count (in seconds) of the executing statement
BUFFER_GETS	NUMBER		Total number of buffer gets for the statement
DISK_READS	NUMBER		Total disk-read I/O count for the statement
ELAPSED_TIME	NUMBER		Total elapsed time (in seconds) of the executing statement
ROWS_PROCESSED	NUMBER		Total number of rows processed by the statement
EXECUTIONS	NUMBER		Total number of times the statement was executed
FIRST_LOAD_TIME	DATE		???
LAST_EXECUTION_DATE	DATE		Date on which the statement was last executed
PRIORITY	NUMBER		Business importance of the statement: ■ 1 - High ■ 2 - Medium ■ 3 - Low
COMMAND_TYPE	NUMBER		Type of the command
STAT_PERIOD	NUMBER		Unused
ACTIVE_STAT_PERIOD	NUMBER		???
SQL_TEXT	CLOB		Text of the SQL statement
PRECOST	NUMBER		Cost of executing the statement in the workload prior to the recommendations
POSTCOST	NUMBER		Cost of executing the statement in the workload after the recommendations
IMPORTANCE	NUMBER		Advisor-calculated importance value
REC_ID	NUMBER		Associated recommendation identifier
VALIDATED	NUMBER		???

See Also: "USER_ADVISOR_SQLA_WK_STMTS" on page 5-99

DBA_ADVISOR_SQLPLANS

DBA_ADVISOR_SQLPLANS displays a list of all SQL execution plans.

Related View

USER_ADVISOR_SQLPLANS displays a list of all SQL execution plans owned by the current user.

Column	Datatype	NULL	Description
TASK_NAME	VARCHAR2(30)		???
TASK_ID	NUMBER(38)	NOT NULL	???
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	The name of the task execution with which this entry (row) is associated
OBJECT_ID	NUMBER(38)	NOT NULL	???
ATTRIBUTE	VARCHAR2(27)		???
STATEMENT_ID	VARCHAR2(30)		???
PLAN_HASH_VALUE	NUMBER	NOT NULL	???
PLAN_ID	NUMBER	NOT NULL	???
TIMESTAMP	DATE		???
REMARKS	VARCHAR2(4000)		???
OPERATION	VARCHAR2(30)		???
OPTIONS	VARCHAR2(255)		???
OBJECT_NODE	VARCHAR2(128)		???
OBJECT_OWNER	VARCHAR2(30)		???
OBJECT_NAME	VARCHAR2(30)		???
OBJECT_ALIAS	VARCHAR2(65)		???
OBJECT_INSTANCE	NUMBER(38)		???
OBJECT_TYPE	VARCHAR2(30)		???
OPTIMIZER	VARCHAR2(255)		???
SEARCH_COLUMNS	NUMBER		???
ID	NUMBER(38)	NOT NULL	???
PARENT_ID	NUMBER(38)		???
DEPTH	NUMBER(38)		???
POSITION	NUMBER(38)		???
COST	NUMBER(38)		???
CARDINALITY	NUMBER(38)		???
BYTES	NUMBER(38)		???
OTHER_TAG	VARCHAR2(255)		???
PARTITION_START	VARCHAR2(255)		???
PARTITION_STOP	VARCHAR2(255)		???
PARTITION_ID	NUMBER(38)		???
OTHER	LONG		???
DISTRIBUTION	VARCHAR2(30)		???
CPU_COST	NUMBER(38)		???
IO_COST	NUMBER(38)		???
TEMP_SPACE	NUMBER(38)		???
ACCESS_PREDICATES	VARCHAR2(4000)		???
FILTER_PREDICATES	VARCHAR2(4000)		???
PROJECTION	VARCHAR2(4000)		???

Column	Datatype	NULL	Description
TIME	NUMBER(38)		???
QBLOCK_NAME	VARCHAR2(30)		???
OTHER_XML	CLOB		???

DBA_ADVISOR_SQLSTATS

DBA_ADVISOR_SQLSTATS displays a list of all SQL compilation and execution statistics.

Related View

USER_ADVISOR_SQLSTATSdisplays a list of all SQL compilation and execution statistics owned by the current user.

Column	Datatype	NULL	Description
TASK_NAME	VARCHAR2(30)		???
TASK_ID	NUMBER(38)	NOT NULL	???
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	The name of the task execution with which this entry (row) is associated
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.
OBJECT_ID	NUMBER(38)	NOT NULL	???
PLAN_ID	NUMBER	NOT NULL	???
SQL_ID	VARCHAR2(13)	NOT NULL	???
PLAN_HASH_VALUE	NUMBER	NOT NULL	???
PARSE_TIME	NUMBER		???
EXECUTE_ELAPSED_TIME	NUMBER		???
ELAPSED_TIME	NUMBER		???
EXECUTE_CPU_TIME	NUMBER		???
BUFFER_GETS	NUMBER		???
DISK_READS	NUMBER		???
DIRECT_WRITES	NUMBER		???
ROWS_PROCESSED	NUMBER		???
FETCHES	NUMBER		???
EXECUTIONS	NUMBER		???
END_OF_FETCH_COUNT	NUMBER		???
OPTIMIZER_COST	NUMBER		???
OTHER	CLOB		???

DBA_ADVISOR_SQLW_JOURNAL

DBA_ADVISOR_SQLW_JOURNAL displays the journal entries for all workload objects in the database.

Related View

USER_ADVISOR_SQLW_JOURNAL displays the journal entries for the workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload
WORKLOAD_ID	NUMBER	NOT NULL	Identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload object
JOURNAL_ENTRY_SEQ	NUMBER	NOT NULL	Sequence number of the journal entry (unique for each workload). The sequence number is used to order the data.
JOURNAL_ENTRY_TYPE	VARCHAR2(12)		Type of the task: ■ FATAL ■ ERROR ■ WARNING ■ INFORMATION ■ INFORMATION[2 | 3 | 4 | 5 | 6]
JOURNAL_ENTRY	VARCHAR2(4000)		Entry in the journal

See Also: "USER_ADVISOR_SQLW_JOURNAL" on page 5-99

DBA_ADVISOR_SQLW_PARAMETERS

DBA_ADVISOR_SQLW_PARAMETERS displays all workload parameters and their current values in the database.

Related View

USER_ADVISOR_SQLW_PARAMETERS displays the workload parameters and their current values owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task or workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload object
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the parameter
PARAMETER_VALUE	VARCHAR2(4000)	NOT NULL	Value of the parameter. Numeric parameter values are converted to a string equivalent. Possible keywords as values: ■ ALL ■ UNLIMITED ■ UNUSED

Column	Datatype	NULL	Description
PARAMETER_TYPE	VARCHAR2(10)		Datatype of the parameter: ■ NUMBER - Numeric value ■ STRING - String value. If the string contains special characters, then it will be enclosed in single quotes. ■ STRINGLIST - Comma-separated list of string elements. If a string element contains a comma or other special characters, then the element will be enclosed in single quotes. ■ TABLE - Single table reference. A reference contains a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported. ■ TABLELIST - List of one or more comma-separated table references. A reference contains a schema name, followed by an optional table name. If the table name is omitted or is the character %, then the table name is interpreted as a wildcard. SQL quoted identifiers are supported.

See Also: "USER_ADVISOR_SQLW_PARAMETERS" on page 5-99

DBA_ADVISOR_SQLW_STMTS

DBA_ADVISOR_SQLW_STMTS displays rows that correspond to all statements in the workload. All columns are guaranteed to be non-null.

Related View

USER_ADVISOR_SQLW_STMTS displays rows that correspond to the statements in the workload owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
SQL_ID	NUMBER	NOT NULL	Generated identifier of the statement
HASH_VALUE	NUMBER		Hash value for the parent statement in the cache
USERNAME	VARCHAR2(30)		Name of the user executing the statement
MODULE	VARCHAR2(64)		Name of the module issuing the statement
ACTION	VARCHAR2(64)		Module action for the statement
CPU_TIME	NUMBER		Total CPU count (in seconds) of the executing statement
BUFFER_GETS	NUMBER		Total number of buffer gets for the statement
DISK_READS	NUMBER		Total disk-read I/O count for the statement
ELAPSED_TIME	NUMBER		Total elapsed time (in seconds) of the executing statement
ROWS_PROCESSED	NUMBER		Total number of rows processed by the statement
EXECUTIONS	NUMBER		Total number of times the statement was executed
OPTIMIZER_COST	NUMBER		Cost of executing the statement in the workload prior to the recommendations
LAST_EXECUTION_DATE	DATE		Date on which the statement was last executed

Column	Datatype	NULL	Description
PRIORITY	NUMBER		Priority of the statement: ■ 1 - High ■ 2 - Medium ■ 3 - Low
COMMAND_TYPE	NUMBER		Type of the command
STAT_PERIOD	NUMBER		Unused
SQL_TEXT	CLOB		Text of the SQL statement
VALID	NUMBER		Indicates whether the statement is valid for analysis: ■ 0 - Statement will not be analyzed by the EXECUTE_TASK procedure. Typically, the statement references one or more tables that do not have valid statistics. To correct this problem, ensure that the tables have valid statistics and execute the RESET_SQLWKLD procedure on the current workload. ■ 1 - Statement is eligible for analysis by the EXECUTE_TASK procedure.

See Also: "USER_ADVISOR_SQLW_STMTS" on page 5-99

DBA_ADVISOR_SQLW_SUM

DBA_ADVISOR_SQLW_SUM displays an aggregated picture of all SQLWkld workload objects in the database.

Related View

USER_ADVISOR_SQLW_SUM displays an aggregated picture of the SQLWkld workload objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Unique name of the workload
DESCRIPTION	VARCHAR2(256)		User-specified description of the workload
CREATE_DATE	DATE	NOT NULL	Date on which the workload object was created
MODIFY_DATE	DATE	NOT NULL	Date of last update to the current workload
NUM_SELECT_STMT	NUMBER		Number of SELECT statements in the workload
NUM_UPDATE_STMT	NUMBER		Number of UPDATE statements in the workload
NUM_DELETE_STMT	NUMBER		Number of DELETE statements in the workload
NUM_INSERT_STMT	NUMBER		Number of INSERT statements in the workload
NUM_MERGE_STMT	NUMBER		Number of MERGE statements in the workload
SOURCE	VARCHAR2(30)		Optional name that identifies the creator of the object
HOW_CREATED	VARCHAR2(30)		Optional object or template on which the object was based
DATA_SOURCE	VARCHAR2(2000)		Workload data source
READ_ONLY	VARCHAR2(5)		Indicates whether or not the workload can be modified or deleted (TRUE) or not (FALSE)

See Also: "USER_ADVISOR_SQLW_SUM" on page 5-100

DBA_ADVISOR_SQLW_TABLES

DBA_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement.

Related View

USER_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER		Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Name of the workload
SQL_ID	NUMBER		Identifier of the statement
TABLE_OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table

See Also: "USER_ADVISOR_SQLW_TABLES" on page 5-100

DBA_ADVISOR_SQLW_TEMPLATES

DBA_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of all SQLWkld template objects in the database.

Related View

USER_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of the SQLWkld template objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the workload object
WORKLOAD_ID	NUMBER	NOT NULL	Unique identifier number of the workload object
WORKLOAD_NAME	VARCHAR2(30)		Unique name of the workload
DESCRIPTION	VARCHAR2(256)		User-specified description of the workload
CREATE_DATE	DATE	NOT NULL	Date on which the workload object was created
MODIFY_DATE	DATE	NOT NULL	Date of last update to the current workload
SOURCE	VARCHAR2(30)		Optional object or template on which the object was based
READ_ONLY	VARCHAR2(5)		Indicates whether the workload template can be modified or deleted (TRUE) or not (FALSE)

See Also: "USER_ADVISOR_SQLW_TEMPLATES" on page 5-100

DBA_ADVISOR_TASKS

DBA_ADVISOR_TASKS displays information about all tasks in the database. The view contains one row for each task. Each task contains a name that is unique to the owner. Task names are just informational and no uniqueness is enforced within any namespace.

Related View

USER_ADVISOR_TASKS displays information about the tasks owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
CREATED	DATE	NOT NULL	Creation date of the task
LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
PARENT_TASK_ID	NUMBER		Identifier of the parent task (if the task was created as a result of the recommendation of another task)
PARENT_RXED_ID	NUMBER		Identifier of the recommendation within the parent task that resulted in the creation of the task
LAST_EXECUTION	VARCHAR2(30)		Name of the current or last task execution
EXECUTION_TYPE	VARCHAR2(30)		Type of the last execution. This information is optional for single-execution tasks.
EXECUTION_TYPE#	NUMBER		???
EXECUTION_DESCRIPTION	VARCHAR2(256)		Optional description of the last execution
EXECUTION_START	DATE		Execution start date and time of the task
EXECUTION_END	DATE		Execution end date and time of the task
STATUS	VARCHAR2(11)		Current operational status of the task: ■ INITIAL - Initial state of the task; no recommendations are present ■ EXECUTING - Task is currently running ■ INTERRUPTED - Task analysis was interrupted by the user. Recommendation data, if present, can be viewed and reported at this time. ■ COMPLETED - Task successfully completed the analysis operation. Recommendation data can be viewed and reported. ■ ERROR - An error occurred during the analysis operation. Recommendations, if present, can be viewed and reported at this time.
STATUS_MESSAGE	VARCHAR2(4000)		Informational message provided by the advisor, regarding the status
PCT_COMPLETION_TIME	NUMBER		Percent completion, in terms of time, of the task when it is executing
PROGRESS_METRIC	NUMBER		Metric that measures the progress of the task in terms of quality. Each advisor may have its own metric.
METRIC_UNITS	VARCHAR2(64)		Unit of the metric used to measure progress
ACTIVITY_COUNTER	NUMBER		Counter that is updated frequently by the advisor, denoting that useful work is being performed
RECOMMENDATION_COUNT	NUMBER		Number of recommendations produced
ERROR_MESSAGE	VARCHAR2(4000)		Informational message or an error message indicating the current operation or condition
SOURCE	VARCHAR2(30)		Optional name that identifies the creator of the task
HOW_CREATED	VARCHAR2(30)		Optional task or template on which the object was based
READ_ONLY	VARCHAR2(5)		Indicates whether the task is read-only (TRUE) or not (FALSE)

Column	Datatype	NULL	Description
SYSTEM_TASK	VARCHAR2(5)		???
ADVISOR_ID	NUMBER	NOT NULL	Unique identifier for the advisor
STATUS#	NUMBER		???

See Also: "USER_ADVISOR_TASKS" on page 5-100

DBA_ADVISOR_TEMPLATES

DBA_ADVISOR_TEMPLATES displays information about all templates in the database.

Related View

USER_ADVISOR_TEMPLATES displays information about the templates owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_ID	NUMBER	NOT NULL	Unique identifier of the task
TASK_NAME	VARCHAR2(30)		Name of the task
DESCRIPTION	VARCHAR2(256)		User-supplied description of the task
ADVISOR_NAME	VARCHAR2(30)		Advisor associated with the task
CREATED	DATE	NOT NULL	Creation date of the task
LAST_MODIFIED	DATE	NOT NULL	Date on which the task was last modified
SOURCE	VARCHAR2(30)		Optional task or template on which the template was based
READ_ONLY	VARCHAR2(5)		Indicates whether the task can be modified or deleted (TRUE) or not (FALSE)

See Also: "USER_ADVISOR_TEMPLATES" on page 5-100

DBA_ADVISOR_USAGE

DBA_ADVISOR_USAGE displays the usage information for each type of advisor in the database.

Column	Datatype	NULL	Description
ADVISOR_ID	NUMBER	NOT NULL	Type of the advisor
LAST_EXEC_TIME	DATE	NOT NULL	Date of the last execution
NUM_EXECS	NUMBER	NOT NULL	Cumulative number of executions

DBA_ALERT_HISTORY

DBA_ALERT_HISTORY describes a time-limited history of alerts which are no longer outstanding.

Column	Datatype	NULL	Description
SEQUENCE_ID	NUMBER	NOT NULL	Alert sequence number
REASON_ID	NUMBER		ID of the alert reason

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object on which the alert was issued
OBJECT_NAME	VARCHAR2(513)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(64)		Object type
REASON	VARCHAR2(4000)		Reason for the alert
TIME_SUGGESTED	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was last updated
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was first produced
SUGGESTED_ACTION	VARCHAR2(4000)		Advice of the recommended action
ADVISOR_NAME	VARCHAR2(30)		Name of the advisor to be invoked for more information
METRIC_VALUE	NUMBER		Value of the related metrics
MESSAGE_TYPE	VARCHAR2(12)		Message type: ▪ Notification ▪ Warning
MESSAGE_GROUP	VARCHAR2(64)		Name of the message group to which the alert belongs
MESSAGE_LEVEL	NUMBER		Severity message level (1 to 32)
HOSTING_CLIENT_ID	VARCHAR2(64)		ID of the client or security group to which the alert relates
MODULE_ID	VARCHAR2(64)		ID of the module that originated the alert
PROCESS_ID	VARCHAR2(128)		Process ID
HOST_ID	VARCHAR2(256)		DNS host name of the originating host
HOST_NW_ADDR	VARCHAR2(256)		IP or other network address of the originating host
INSTANCE_NAME	VARCHAR2(16)		Originating instance name
INSTANCE_NUMBER	NUMBER		Originating instance number
USER_ID	VARCHAR2(30)		User ID
EXECUTION_CONTEXT_ID	VARCHAR2(60)		ID of the thread of execution
ERROR_INSTANCE_ID	VARCHAR2(142)		ID of an error instance plus a sequence number
RESOLUTION	VARCHAR2(7)		Resolution: ▪ cleared ▪ N/A

DBA_ALL_TABLES

DBA_ALL_TABLES describes all object tables and relational tables in the database. Its columns are the same as those in ALL_ALL_TABLES.

See Also: "ALL_ALL_TABLES" on page 2-5

DBA_APPLICATION_ROLES

DBA_APPLICATION_ROLES describes all the roles that have authentication policy functions defined.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the application role
SCHEMA	VARCHAR2(30)	NOT NULL	Schema of the authorized package

Column	Datatype	NULL	Description
PACKAGE	VARCHAR2(30)	NOT NULL	Name of the authorized package

DBA_APPLY

DBA_APPLY displays information about all apply processes in the database. Its columns are the same as those in ALL_APPLY.

See Also: "ALL_APPLY" on page 2-8

DBA_APPLY_CONFLICT_COLUMNS

DBA_APPLY_CONFLICT_COLUMNS displays information about the conflict handlers on all tables in the database. Its columns are the same as those in ALL_APPLY_CONFLICT_COLUMNS.

See Also: "ALL_APPLY_CONFLICT_COLUMNS" on page 2-9

DBA_APPLY_DML_HANDLERS

DBA_APPLY_DML_HANDLERS displays information about the DML handlers on all tables in the database. Its columns are the same as those in ALL_APPLY_DML_HANDLERS.

See Also: "ALL_APPLY_DML_HANDLERS" on page 2-9

DBA_APPLY_ENQUEUE

DBA_APPLY_ENQUEUE displays information about the apply enqueue actions for all rules in the database. Its columns are the same as those in ALL_APPLY_ENQUEUE.

See Also: "ALL_APPLY_ENQUEUE" on page 2-10

DBA_APPLY_ERROR

DBA_APPLY_ERROR displays information about the error transactions generated by all apply processes in the database. Its columns are the same as those in ALL_APPLY_ERROR.

See Also: "ALL_APPLY_ERROR" on page 2-10

DBA_APPLY_EXECUTE

DBA_APPLY_EXECUTE displays information about the apply execute actions for all rules in the database. Its columns are the same as those in ALL_APPLY_EXECUTE.

See Also: "ALL_APPLY_EXECUTE" on page 2-11

DBA_APPLY_INSTANTIATED_GLOBAL

DBA_APPLY_INSTANTIATED_GLOBAL displays information about databases for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database that was instantiated
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the database. Only changes committed after this SCN are applied by an apply process.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_INSTANTIATED_OBJECTS

DBA_APPLY_INSTANTIATED_OBJECTS displays information about objects for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database where the object originated
SOURCE_OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object at the source database
SOURCE_OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object at the source database
SOURCE_OBJECT_TYPE	VARCHAR2(11)		Type of the object at the source database
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the object. Only changes committed after this SCN are applied by an apply process.
IGNORE_SCN	NUMBER		SCN below which the instantiation SCN cannot be set. This value corresponds to the SCN value at the source database at the time when the object was prepared for instantiation.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_INSTANTIATED_SCHEMAS

DBA_APPLY_INSTANTIATED_SCHEMAS displays information about schemas for which an instantiation SCN has been set.

Column	Datatype	NULL	Description
SOURCE_DATABASE	VARCHAR2(128)	NOT NULL	Name of the database where the schema originated
SOURCE_SCHEMA	VARCHAR2(30)		Name of the schema at the source database
INSTANTIATION_SCN	NUMBER		Instantiation SCN for the schema. Only changes committed after this SCN are applied by an apply process.
APPLY_DATABASE_LINK	VARCHAR2(128)		Database link to which changes are applied. If null, then changes are applied to the local database.

DBA_APPLY_KEY_COLUMNS

DBA_APPLY_KEY_COLUMNS displays information about the substitute key columns for all tables in the database. Its columns are the same as those in ALL_APPLY_KEY_COLUMNS.

See Also: "ALL_APPLY_KEY_COLUMNS" on page 2-11

DBA_APPLY_OBJECT_DEPENDENCIES

DBA_APPLY_OBJECT_DEPENDENCIES displays information about the object dependencies for all apply processes in the database.

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
PARENT_OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Parent of the object owner
PARENT_OBJECT_NAME	VARCHAR2(30)	NOT NULL	Parent of the named object

DBA_APPLY_PARAMETERS

DBA_APPLY_PARAMETERS displays information about the parameters for all apply processes in the database. Its columns are the same as those in ALL_APPLY_PARAMETERS.

See Also: "ALL_APPLY_PARAMETERS" on page 2-12

DBA_APPLY_PROGRESS

DBA_APPLY_PROGRESS displays information about the progress made by all apply processes in the database. Its columns are the same as those in ALL_APPLY_PROGRESS.

See Also: "ALL_APPLY_PROGRESS" on page 2-12

DBA_APPLY_SPILL_TXN

DBA_APPLY_SPILL_TXN displays information about the transactions spilled from memory to hard disk by all apply processes in the database.

Column	Datatype	NULL	Description
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process that spilled one or more transactions
XIDUSN	NUMBER	NOT NULL	Transaction ID undo segment number
XIDSLT	NUMBER	NOT NULL	Transaction ID slot number
XIDSQN	NUMBER	NOT NULL	Transaction ID sequence number
FIRST_SCN	NUMBER	NOT NULL	SCN of the first message in the transaction
MESSAGE_COUNT	NUMBER		Number of messages spilled for the transaction
FIRST_MESSAGE_CREATE_TIME	DATE		Source creation time of the first message in the transaction
SPILL_CREATION_TIME	DATE		Time the first message was spilled

DBA_APPLY_TABLE_COLUMNS

DBA_APPLY_TABLE_COLUMNS displays information about the destination table object columns for all tables in the database. Its columns are the same as those in ALL_APPLY_TABLE_COLUMNS.

See Also: "ALL_APPLY_TABLE_COLUMNS" on page 2-13

DBA_APPLY_VALUE_DEPENDENCIES

DBA_APPLY_VALUE_DEPENDENCIES displays information about the value dependencies for all apply processes in the database.

Column	Datatype	NULL	Description
DEPENDENCY_NAME	VARCHAR2(30)	NOT NULL	Name of the dependency
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
COLUMN_POSITION	NUMBER		Position of the column

DBA_AQ_AGENT_PRIVS

DBA_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to all users in the database.

Related View

USER_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to the current user. This view does not display the DB_USERNAME column.

Column	Datatype	NULL	Description
AGENT_NAME	VARCHAR2(30)	NOT NULL	Name of the AQ agent
DB_USERNAME	VARCHAR2(30)		Name of the database user that the agent maps to
HTTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through HTTP (YES) or not (NO)
SMTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through SMTP (YES) or not (NO)

See Also: "USER_AQ_AGENT_PRIVS" on page 5-100

DBA_AQ_AGENTS

DBA_AQ_AGENTS displays information about all registered AQ agents in the database.

Column	Datatype	NULL	Description
AGENT_NAME	VARCHAR2(30)	NOT NULL	Name of the AQ agent
HTTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through HTTP (YES) or not (NO)
SMTP_ENABLED	VARCHAR2(4)		Indicates whether the agent is allowed to access AQ through SMTP (YES) or not (NO)

DBA_ARGUMENTS

DBA_ARGUMENTS lists the arguments of the procedures and functions that are available in the database. Its columns are the same as those in ALL_ARGUMENTS.

See Also: "ALL_ARGUMENTS" on page 2-13

DBA_ASSEMBLIES

DBA_ASSEMBLIES provides information about all assemblies in the database. Its columns are the same as those in ALL_ASSEMBLIES.

See Also: "ALL_ASSEMBLIES" on page 2-15

DBA_ASSOCIATIONS

DBA_ASSOCIATIONS describes all user-defined statistics in the database. Its columns are the same as those in ALL_ASSOCIATIONS.

See Also: "ALL_ASSOCIATIONS" on page 2-16

DBA_ATTRIBUTE_TRANSFORMATIONS

DBA_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for all transformations in the database.

Related View

USER_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for the transformations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TRANSFORMATION_ID	NUMBER	NOT NULL	Unique identifier for the transformation
OWNER	VARCHAR2(30)	NOT NULL	Owning user of the transformation
NAME	VARCHAR2(30)	NOT NULL	Transformation name
FROM_TYPE	VARCHAR2(61)		Source type name
TO_TYPE	VARCHAR2(91)		Target type name
ATTRIBUTE	NUMBER	NOT NULL	Target type attribute number
ATTRIBUTE_TRANSFORMATION	VARCHAR2(4000)		Transformation function for the attribute

See Also: "USER_ATTRIBUTE_TRANSFORMATIONS" on page 5-101

DBA_AUDIT_EXISTS

DBA_AUDIT_EXISTS displays audit trail entries produced by AUDIT EXISTS and AUDIT NOT EXISTS.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Intended creator of the non-existent object
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: ▪ 0 - Action succeeded ▪ 2004 - Security violation
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_EDITION	VARCHAR2(30)		???

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: "AUDIT_TRAIL" on page 1-24

DBA_AUDIT_OBJECT

DBA_AUDIT_OBJECT displays audit trail records for all objects in the database.

Related View

USER_AUDIT_OBJECT displays audit trail records for all objects accessible to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both).
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: ▪ 0 - Action succeeded ▪ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_EDITION	VARCHAR2(30)		???

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: "AUDIT_TRAIL" on page 1-24

DBA_AUDIT_POLICIES

DBA_AUDIT_POLICIES describes all fine-grained auditing policies in the database. Its columns are the same as those in ALL_AUDIT_POLICIES.

See Also: "ALL_AUDIT_POLICIES" on page 2-16

DBA_AUDIT_POLICY_COLUMNS

DBA_AUDIT_POLICY_COLUMNS describes all fine-grained auditing policy columns in the database. Its columns are the same as those in ALL_AUDIT_POLICY_COLUMNS.

See Also: "ALL_AUDIT_POLICY_COLUMNS" on page 2-17

DBA_AUDIT_SESSION

DBA_AUDIT_SESSION displays all audit trail records concerning CONNECT and DISCONNECT.

Related View

USER_AUDIT_SESSION contains audit trail records concerning connections and disconnections of the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone

Column	Datatype	NULL	Description
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL
LOGOFF_TIME	DATE		Date and time of user log off
LOGOFF_LREAD	NUMBER		Logical reads for the session
LOGOFF_PREAD	NUMBER		Physical reads for the session
LOGOFF_LWRITE	NUMBER		Logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Deadlocks detected during the session
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: ■ 0 - Action succeeded ■ 2004 - Security violation
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process

DBA_AUDIT_STATEMENT

DBA_AUDIT_STATEMENT displays audit trail records concerning GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM statements throughout the database.

Related View

USER_AUDIT_STATEMENT displays audit trail records for the same statements issued by the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column in DBA_AUDIT_TRAIL

Column	Datatype	NULL	Description
NEW_NAME	VARCHAR2(128)		New name of an object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Signifies the role or system privilege was granted with the ADMIN option
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: - for none, S for success, F for failure, and B for both).
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail, inserted by the application
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: ■ 0 - Action succeeded ■ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_EDITION	VARCHAR2(30)		???

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: "AUDIT_TRAIL" on page 1-24

DBA_AUDIT_TRAIL

DBA_AUDIT_TRAIL displays all audit trail entries.

Related View

USER_AUDIT_TRAIL displays all audit trail entries related to the current user.

Column	Datatype	NULL	Description
OS_USERNAME	VARCHAR2(255)		Operating system login username of the user whose actions were audited
USERNAME	VARCHAR2(30)		Name (not ID number) of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
TIMESTAMP	DATE		Date and time of the creation of the audit trail entry (date and time of user login for entries created by AUDIT SESSION) in the local database session time zone
OWNER	VARCHAR2(30)		Creator of the object affected by the action
OBJ_NAME	VARCHAR2(128)		Name of the object affected by the action
ACTION	NUMBER	NOT NULL	Numeric action type code. The corresponding name of the action type is in the ACTION_NAME column.
ACTION_NAME	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of the object after a RENAME or the name of the underlying object
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Indicates whether the role or system privilege was granted with the ADMIN option
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: ■ - - None ■ S - Success ■ F - Failure ■ B - Both
LOGOFF_TIME	DATE		Date and time of user log off

Column	Datatype	NULL	Description
LOGOFF_LREAD	NUMBER		Logical reads for the session
LOGOFF_PREAD	NUMBER		Physical reads for the session
LOGOFF_LWRITE	NUMBER		Logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Deadlocks detected during the session
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail entry, providing more information about the statement audited Also indicates how the user was authenticated. The method can be one of the following: ■ DATABASE - Authentication was done by password ■ NETWORK - Authentication was done by Oracle Net Services or the Advanced Security option ■ PROXY - Client was authenticated by another user; the name of the proxy user follows the method type
SESSIONID	NUMBER	NOT NULL	Numeric ID for each Oracle session
ENTRYID	NUMBER	NOT NULL	Numeric ID for each audit trail entry in the session
STATEMENTID	NUMBER	NOT NULL	Numeric ID for each statement run
RETURNCODE	NUMBER	NOT NULL	Oracle error code generated by the action. Some useful values: ■ 0 - Action succeeded ■ 2004 - Security violation
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
SESSION_CPU	NUMBER		Amount of CPU time used by each Oracle session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the creation of the audit trail entry (timestamp of user login for entries created by AUDIT SESSION) in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
SCN	NUMBER		System change number (SCN) of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_EDITION	VARCHAR2(30)		???

Note: The SQL_BIND and SQL_TEXT columns are only populated if the AUDIT_TRAIL initialization parameter is set to db_extended.

See Also: "AUDIT_TRAIL" on page 1-24

DBA_AUTO_SEGADV_CTL

DBA_AUTO_SEGADV_CTL exposes the control information used by the segment advisor. This information gives the DBA an idea of what is happening in the auto advisor.

Column	Datatype	NULL	Description
AUTO_TASKID	NUMBER		Unique task ID generated by the auto advisor
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
SEGMENT_OWNER	VARCHAR2(30)		Owner of the segment
SEGMENT_NAME	VARCHAR2(81)		Name of the segment
SEGMENT_TYPE	VARCHAR2(18)		The type of segment (TABLE, INDEX, and so on)
PARTITION_NAME	VARCHAR2(30)		Name of the subsegment (if partitioned)
STATUS	VARCHAR2(40)		Status of the analysis: ■ NEW - the segment/tablespace has not been analyzed ■ BEING-PROCESSED - the segment/tablespace is being processed ■ COMPLETE - the segment/tablespace has been analyzed ■ ERROR - problem with the segment/tablespace
REASON	VARCHAR2(40)		Reason why this segment was chosen
REASON_VALUE	NUMBER		A value associated with the reason
CREATION_TIME	TIMESTAMP(6)		Time when this entry was created
PROCESSED_TASKID	NUMBER		Auto advisor task that was used to process the segment/tablespace
END_TIME	TIMESTAMP(6)		Time at which the advisor task was completed

DBA_AUTO_SEGADV_SUMMARY

DBA_AUTO_SEGADV_SUMMARY provides a summary of the auto advisor task runs.

Column	Datatype	NULL	Description
AUTO_TASKID	NUMBER	NOT NULL	Unique auto task ID
SNAPID	NUMBER		Maximum AWR snapid used to process the segments
SEGMENTS_SELECTED	NUMBER		Number of segments chosen for analysis
SEGMENTS_PROCESSED	NUMBER		Number of segments actually processed
TABLESPACE_SELECTED	NUMBER		Number of tablespaces chosen for analysis
TABLESPACE_PROCESSED	NUMBER		Number of tablespaces actually processed
RECOMMENDATIONS_COUNT	NUMBER		Number of recommendations generated
START_TIME	TIMESTAMP(6)		Time at which the auto task was started
END_TIME	TIMESTAMP(6)		Time at which the auto task ended

DBA_AUTOTASK_CLIENT

DBA_AUTOTASK_CLIENT displays statistical data for each automated maintenance task over 7-day and 30-day periods.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the client
STATUS	VARCHAR2(8)		Job status: ■ ENABLED ■ DISABLED
CONSUMER_GROUP	VARCHAR2(30)		Consumer group used for normal priority jobs for this client
CLIENT_TAG	VARCHAR2(2)		Tag used to identify jobs for this client
PRIORITY_OVERRIDE	VARCHAR2(7)		User-specified priority at which the task executes: ■ URGENT ■ HIGH ■ MEDIUM ■ LOW
ATTRIBUTES	VARCHAR2(4000)		Attributes of the client
WINDOW_GROUP	VARCHAR2(64)		Window group used to schedule jobs
SERVICE_NAME	VARCHAR2(64)		Name of the service on which jobs will execute for the client
RESOURCE_PERCENTAGE	NUMBER		Percentage of maintenance resources for high priority maintenance tasks for this client
USE_RESOURCE_ESTIMATES	VARCHAR2(5)		Indicates whether resource estimates are used for this client (TRUE) or not (FALSE)
MEAN_JOB_DURATION	INTERVAL DAY(9) TO SECOND(9)		Average elapsed time for a job for this client (in seconds)
MEAN_JOB_CPU	INTERVAL DAY(9) TO SECOND(9)		Average CPU time for a job submitted by this client (in seconds)
MEAN_JOB_ATTEMPTS	NUMBER		Average number of attempts it takes to complete a task
MEAN_INCOMING_TASKS_7_DAYS	NUMBER		Average number of incoming tasks at the Maintenance Window Start over the last 7 days
MEAN_INCOMING_TASKS_30_DAYS	NUMBER		Average number of incoming tasks at the Maintenance Window Start over the last 30 days
TOTAL_CPU_LAST_7_DAYS	INTERVAL DAY(9) TO SECOND(9)		Cumulative CPU time used by the jobs for this client over the last 7 days (in seconds)
TOTAL_CPU_LAST_30_DAYS	INTERVAL DAY(9) TO SECOND(9)		Cumulative CPU time used by the jobs for this client over the last 30 days (in seconds)
MAX_DURATION_LAST_7_DAYS	INTERVAL DAY(3) TO SECOND(0)		Maximum elapsed time for a job over the last 7 days (in seconds)
MAX_DURATION_LAST_30_DAYS	INTERVAL DAY(3) TO SECOND(0)		Maximum elapsed time for a job over the last 30 days (in seconds)
WINDOW_DURATION_LAST_7_DAYS	INTERVAL DAY(9) TO SECOND(9)		Total time during which the client was active during the last 7 days
WINDOW_DURATION_LAST_30_DAYS	INTERVAL DAY(9) TO SECOND(9)		Total time during which the client was active during the last 30 days

DBA_AUTOTASK_CLIENT_HISTORY

DBA_AUTOTASK_CLIENT_HISTORY displays per-window history of job execution counts for each automated maintenance task. This information is viewable in the Job History page of Enterprise Manager.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the client

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(65)		Name of the maintenance window
WINDOW_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Maintenance window start time
WINDOW_DURATION	INTERVAL DAY(9) TO SECOND(6)		Window duration (NULL for currently open window)
JOBS_CREATED	NUMBER		Number of jobs created on behalf of the client in this window
JOBS_STARTED	NUMBER		Number of jobs started on behalf of the client during the maintenance window
JOBS_COMPLETED	NUMBER		Number of jobs successfully completed on behalf of the client during the maintenance window
WINDOW_END_TIME	TIMESTAMP(6) WITH TIME ZONE		Window end time

DBA_AUTOTASK_CLIENT_JOB

DBA_AUTOTASK_CLIENT_JOB displays information about currently running Scheduler jobs created for automated maintenance tasks. It provides information about some objects targeted by those jobs, as well as some additional statistics from previous instantiations of the same task. Some of this additional data is taken from generic Scheduler views.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the client
JOB_NAME	VARCHAR2(65)		Name of the job
JOB_SCHEDULER_STATUS	VARCHAR2(15)		Job status: ▪ DISABLED ▪ RETRY SCHEDULED ▪ SCHEDULED ▪ RUNNING ▪ COMPLETED ▪ BROKEN ▪ FAILED ▪ REMOTE ▪ SUCCEEDED ▪ CHAIN_STALLED
TASK_NAME	VARCHAR2(64)		Name of the task being performed
TASK_TARGET_TYPE	VARCHAR2(64)		Type of the target being processed
TASK_TARGET_NAME	VARCHAR2(513)	NOT NULL	Name of the target
TASK_PRIORITY	VARCHAR2(7)		Priority of the task: ▪ URGENT ▪ HIGH ▪ MEDIUM ▪ LOW
TASK_OPERATION	VARCHAR2(64)		Operation performed on the object

DBA_AUTOTASK_JOB_HISTORY

DBA_AUTOTASK_JOB_HISTORY displays the history of automated maintenance task job runs. Jobs are added to this view after they finish executing.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the automated maintenance client
WINDOW_NAME	VARCHAR2(65)		Name of the maintenance window
WINDOW_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Start time of the maintenance window
WINDOW_DURATION	INTERVAL DAY(9) TO SECOND(6)		Duration of the maintenance window
JOB_NAME	VARCHAR2(65)		Name of the maintenance job
JOB_STATUS	VARCHAR2(30)		Status of the maintenance job
JOB_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Start time of the maintenance job
JOB_DURATION	INTERVAL DAY(3) TO SECOND(0)		Duration of the maintenance job
JOB_ERROR	NUMBER		Error code for the job (if any)
JOB_INFO	VARCHAR2(4000)		Additional information about the job

DBA_AUTOTASK_OPERATION

DBA_AUTOTASK_OPERATION displays all automated maintenance task operations for each client.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the client
OPERATION_NAME	VARCHAR2(64)		Name of the operation
OPERATION_TAG	VARCHAR2(3)		Tag for the operation
PRIORITY_OVERRIDE	VARCHAR2(7)		User-specified priority at which the task executes: ▪ URGENT ▪ HIGH ▪ MEDIUM ▪ LOW
ATTRIBUTES	VARCHAR2(4000)		Attributes of the operation
USE_RESOURCE_ESTIMATES	VARCHAR2(5)		Indicates whether resource usage estimates are used for the operation (TRUE) or not (FALSE)
STATUS	VARCHAR2(8)		Job status: ▪ ENABLED ▪ DISABLED

DBA_AUTOTASK_SCHEDULE

DBA_AUTOTASK_SCHEDULE displays the schedule of maintenance windows for the next 32 days for each client.

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(30)		Name of the maintenance window
START_TIME	TIMESTAMP(6) WITH TIME ZONE		Projected start time of the window
DURATION	INTERVAL DAY(3) TO SECOND(0)		Currently defined duration of the window (NULL for the currently open window)

DBA_AUTOTASK_TASK

DBA_AUTOTASK_TASK displays information about current and past automated maintenance tasks.

Column	Datatype	NULL	Description
CLIENT_NAME	VARCHAR2(64)		Name of the client
TASK_NAME	VARCHAR2(64)		Name of the task being performed
TASK_TARGET_TYPE	VARCHAR2(64)		Target type of the task
TASK_TARGET_NAME	VARCHAR2(513)	NOT NULL	Name of the target
OPERATION_NAME	VARCHAR2(64)		Operation performed on the object
ATTRIBUTES	VARCHAR2(4000)		Attributes of the task
TASK_PRIORITY	NUMBER		Task priority, relative to other tasks for this Client
PRIORITY_OVERRIDE	NUMBER		Task priority as overridden by the user
STATUS	VARCHAR2(8)		Status of the task: ■ DISABLED ■ DEFERRED ■ ENABLED
DEFERRED_WINDOW_NAME	VARCHAR2(65)		Appropriate window for this task
CURRENT_JOB_NAME	VARCHAR2(65)		Name of the currently scheduled job, if any
JOB_SCHEDULER_STATUS	VARCHAR2(15)		Job status: ■ DISABLED ■ RETRY SCHEDULED ■ SCHEDULED ■ RUNNING ■ COMPLETED ■ BROKEN ■ FAILED ■ REMOTE ■ SUCCEEDED ■ CHAIN_STALLED
ESTIMATE_TYPE	VARCHAR2(7)		Type of resource estimates applied: ■ DERIVED ■ FORCED ■ LOCKED
ESTIMATED_WEIGHT	NUMBER		Task weight indicator
ESTIMATED_DURATION	NUMBER		Estimated elapsed time for the job (in seconds)
ESTIMATED_CPU_TIME	NUMBER		Estimated CPU time for the job (in seconds)
ESTIMATED_TEMP	NUMBER		Estimated temporary space usage for the job (in KB)
ESTIMATED_DOP	NUMBER		Estimated degree of parallelism for the job
ESTIMATED_IO_RATE	NUMBER		Estimated I/O utilization for the job (in KB per second)
ESTIMATED_UNDO_RATE	NUMBER		Estimated UNDO generation rate for the job (in KB per second)
RETRY_COUNT	NUMBER		Number of attempts to perform this task since the last successful attempt
LAST_GOOD_DATE	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the last successful attempt to perform this task
LAST_GOOD_PRIORITY	NUMBER		Job priority of the last successful attempt to perform this task

Column	Datatype	NULL	Description
LAST_GOOD_DURATION	NUMBER		Elapsed time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_CPU_TIME	NUMBER		CPU time for the job (in seconds) of the last successful attempt to perform this task
LAST_GOOD_TEMP	NUMBER		Temporary space usage for the job (in KB) of the last successful attempt to perform this task
LAST_GOOD_DOP	NUMBER		Peak degree of parallelism for the job during the last successful attempt to perform this task
LAST_GOOD_IO_RATE	NUMBER		I/O utilization rate for the job (in KB per second) of the last successful attempt to perform this task
LAST_GOOD_UNDO_RATE	NUMBER		NDO generation rate (in KB per second) of the last successful attempt to perform this task
LAST_GOOD_CPU_WAIT	NUMBER		Resource Manager wait time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_IO_WAIT	NUMBER		Resource Manager wait time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_UNDO_WAIT	NUMBER		Resource Manager wait time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_TEMP_WAIT	NUMBER		Resource Manager wait time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_CONCURRENCY	NUMBER		Concurrency wait time (in seconds) of the last successful attempt to perform this task
LAST_GOOD_CONTENTION	NUMBER		Contention wait time (in seconds) of the last successful attempt to perform this task
NEXT_TRY_DATE	TIMESTAMP(6) WITH TIME ZONE		Next projected start time for the deferred maintenance window
LAST_TRY_DATE	TIMESTAMP(6) WITH TIME ZONE		Time at which the task was last attempted
LAST_TRY_PRIORITY	NUMBER		Task priority at the time of the last attempt
LAST_TRY_RESULT	VARCHAR2(36)		Result code of the last execution of the task: ▪ SUCCEEDED ▪ FAILED ▪ STOPPED BY USER ACTION ▪ STOPPED AT END OF MAINTENANCE WINDOW ▪ STOPPED AT INSTANCE SHUTDOWN ▪ STOPPED
LAST_TRY_DURATION	NUMBER		Elapsed time of the last run (in seconds)
LAST_TRY_CPU_TIME	NUMBER		CPU time during the last run (in seconds)
LAST_TRY_TEMP	NUMBER		Temporary space usage for the job (in KB) for the last run
LAST_TRY_DOP	NUMBER		Peak degree of parallelism for the job during the last run
LAST_TRY_IO_RATE	NUMBER		I/O rate during the last run (in seconds)
LAST_TRY_UNDO_RATE	NUMBER		UNDO generation rate during the last run (in seconds)
LAST_TRY_CPU_WAIT	NUMBER		Time spent waiting for CPU during the last run (in seconds)
LAST_TRY_IO_WAIT	NUMBER		Time spent waiting for I/O during the last run (in seconds)
LAST_TRY_UNDO_WAIT	NUMBER		Time spent waiting for UNDO during the last run (in seconds)
LAST_TRY_TEMP_WAIT	NUMBER		Time spent waiting for temporary space during the last run (in seconds)

Column	Datatype	NULL	Description
LAST_TRY_CONCURRENCY	NUMBER		Concurrency wait time during the last run (in seconds)
LAST_TRY_CONTENTION	NUMBER		Contention wait time during the last run (in seconds)
MEAN_GOOD_DURATION	NUMBER		Average elapsed time for successful executions of this task (in seconds)
MEAN_GOOD_CPU_TIME	NUMBER		Average CPU time for successful executions of this task (in seconds)
MEAN_GOOD_TEMP	NUMBER		Average temporary space usage for successful executions of this task (in KB)
MEAN_GOOD_DOP	NUMBER		Average peak degree of parallelism for successful executions of this task
MEAN_GOOD_IO	NUMBER		Average I/O utilization for successful executions of this task (in KB per second)
MEAN_GOOD_UNDO	NUMBER		Average UNDO generation rate for this task (in KB per second)
MEAN_GOOD_CPU_WAIT	NUMBER		Average time waiting for CPU for successful executions of this task (in seconds)
MEAN_GOOD_IO_WAIT	NUMBER		Average time waiting for I/O for successful executions of this task (in seconds)
MEAN_GOOD_UNDO_WAIT	NUMBER		Average time waiting for UNDO for successful executions of this task (in seconds)
MEAN_GOOD_TEMP_WAIT	NUMBER		Average time waiting for temporary space for successful executions of this task (in seconds)
MEAN_GOOD_CONCURRENCY	NUMBER		Average concurrency wait time for successful executions of this task (in seconds)
MEAN_GOOD_CONTENTION	NUMBER		Average contention wait time for successful executions of this task (in seconds)
INFO_FIELD_1	VARCHAR2(4000)		Client-interpreted information
INFO_FIELD_2	CLOB		Client-interpreted information
INFO_FIELD_3	NUMBER		Client-interpreted information
INFO_FIELD_4	NUMBER		Client-interpreted information

DBA_AUTOTASK_WINDOW_CLIENTS

DBA_AUTOTASK_WINDOW_CLIENTS displays the windows that belong to MAINTENANCE_WINDOW_GROUP, along with the Enabled or Disabled status for the window for each maintenance task. Primarily used by Enterprise Manager.

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(30)	NOT NULL	Name of the maintenance window
WINDOW_NEXT_TIME	TIMESTAMP(6) WITH TIME ZONE		Next scheduled window open time unless the window is disabled
WINDOW_ACTIVE	VARCHAR2(5)		Indicates whether the window is currently active (open) (TRUE) or not (FALSE)
AUTOTASK_STATUS	VARCHAR2(8)		Status of the automated maintenance task subsystem: ■ ENABLED ■ DISABLED
OPTIMIZER_STATS	VARCHAR2(8)		Status of optimizer statistics gathering: ■ ENABLED ■ DISABLED

Column	Datatype	NULL	Description
SEGMENT_ADVISOR	VARCHAR2(8)		Status of Segment Advisor: ■ ENABLED ■ DISABLED
SQL_TUNE_ADVISOR	VARCHAR2(8)		Status of SQL Tuning Advisor: ■ ENABLED ■ DISABLED
HEALTH_MONITOR	VARCHAR2(8)		Status of Health Monitor: ■ ENABLED ■ DISABLED

DBA_AUTOTASK_WINDOW_HISTORY

DBA_AUTOTASK_WINDOW_HISTORY displays historical information for automated maintenance task windows.

Column	Datatype	NULL	Description
WINDOW_NAME	VARCHAR2(65)		Name of the maintenance window
WINDOW_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Window start time
WINDOW_END_TIME	TIMESTAMP(6) WITH TIME ZONE		Window end time

DBA_AW_PS

DBA_AW_PS describes the page spaces in all analytic workspaces in the database. Its columns are the same as those in ALL_AW_PS.

See Also: "ALL_AW_PS" on page 2-18

DBA_AWS

DBA_AWS describes all analytic workspaces in the database. Its columns are the same as those in ALL_AWS.

See Also: "ALL_AWS" on page 2-18

DBA_BASE_TABLE_MVIEWS

DBA_BASE_TABLE_MVIEWS describes all materialized views using materialized view logs in the database. Its columns are the same as those in ALL_BASE_TABLE_MVIEWS.

See Also: "ALL_BASE_TABLE_MVIEWS" on page 2-19

DBA_BLOCKERS

DBA_BLOCKERS displays a session if it is not waiting for a locked object but is holding a lock on an object for which another session is waiting.

Column	Datatype	NULL	Description
HOLDING_SESSION	NUMBER		Session holding a lock

DBA_CAPTURE

`DBA_CAPTURE` displays information about all capture processes in the database. Its columns are the same as those in `ALL_CAPTURE`.

> **See Also:** "ALL_CAPTURE" on page 2-19

DBA_CAPTURE_EXTRA_ATTRIBUTES

`DBA_CAPTURE_EXTRA_ATTRIBUTES` displays information about the extra attributes for all capture processes in the database. Its columns are the same as those in `ALL_CAPTURE_EXTRA_ATTRIBUTES`.

> **See Also:** "ALL_CAPTURE_EXTRA_ATTRIBUTES" on page 2-21

DBA_CAPTURE_PARAMETERS

`DBA_CAPTURE_PARAMETERS` displays information about the parameters for all capture processes in the database. Its columns are the same as those in `ALL_CAPTURE_PARAMETERS`.

> **See Also:** "ALL_CAPTURE_PARAMETERS" on page 2-21

DBA_CAPTURE_PREPARED_DATABASE

`DBA_CAPTURE_PREPARED_DATABASE` displays information about when the local database was prepared for instantiation. Its columns are the same as those in `ALL_CAPTURE_PREPARED_DATABASE`.

> **See Also:**
>
> - "ALL_CAPTURE_PREPARED_DATABASE" on page 2-22
> - *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

DBA_CAPTURE_PREPARED_SCHEMAS

`DBA_CAPTURE_PREPARED_SCHEMAS` displays information about all schemas prepared for instantiation at the local database. Its columns are the same as those in `ALL_CAPTURE_PREPARED_SCHEMAS`.

> **See Also:**
>
> - "ALL_CAPTURE_PREPARED_SCHEMAS" on page 2-22
> - *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

DBA_CAPTURE_PREPARED_TABLES

`DBA_CAPTURE_PREPARED_TABLES` displays information about all tables prepared for instantiation at the local database. Its columns are the same as those in `ALL_CAPTURE_PREPARED_TABLES`.

See Also:

- "ALL_CAPTURE_PREPARED_TABLES" on page 2-23
- *Oracle Streams Replication Administrator's Guide* for more information about the possible values for the supplemental logging columns in this view

DBA_CATALOG

DBA_CATALOG lists all tables, views, clusters, synonyms, and sequences in the database. Its columns are the same as those in "ALL_CATALOG" on page 2-24.

DBA_CHANGE_NOTIFICATION_REGS

DBA_CHANGE_NOTIFICATION_REGS describes all change notification registrations in the database.

Related View

USER_CHANGE_NOTIFICATION_REGS describes the change notification registrations owned by the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(31)		Owner of the registration
REGID	NUMBER		Internal registration ID
REGFLAGS	NUMBER		Registration flags
CALLBACK	VARCHAR2(256)		Notification callback
OPERATIONS_FILTER	NUMBER		Operations filter (if specified)
CHANGELAG	NUMBER		Transaction lag between notifications (if specified)
TIMEOUT	NUMBER		Registration timeout (if specified)
TABLE_NAME	VARCHAR2(63)		Name of the registered table

See Also: "USER_CHANGE_NOTIFICATION_REGS" on page 5-102

DBA_CLU_COLUMNS

DBA_CLU_COLUMNS maps all table columns to related cluster columns.

Related View

USER_CLU_COLUMNS maps all table columns owned by the current user to related cluster columns. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the cluster
CLUSTER_NAME	VARCHAR2(30)	NOT NULL	Cluster name
CLU_COLUMN_NAME	VARCHAR2(30)	NOT NULL	Key column in the cluster
TABLE_NAME	VARCHAR2(30)	NOT NULL	Clustered table name
TAB_COLUMN_NAME	VARCHAR2(4000)		Key column or attribute of the object type column

DBA_CLUSTER_HASH_EXPRESSIONS

DBA_CLUSTER_HASH_EXPRESSIONS lists hash functions for all hash clusters in the database. Its columns are the same as those in "ALL_CLUSTER_HASH_EXPRESSIONS" on page 2-27.

DBA_CLUSTERS

DBA_CLUSTERS describes all clusters in the database. Its columns are the same as those in "ALL_CLUSTERS" on page 2-27.

DBA_COL_COMMENTS

DBA_COL_COMMENTS displays comments on the columns of all tables and views in the database. Its columns are the same as those in ALL_COL_COMMENTS.

See Also: "ALL_COL_COMMENTS" on page 2-28

DBA_COL_PENDING_STATS

DBA_COL_PENDING_STATS describes the pending statistics of all columns in the database. Its columns are the same as those in ALL_COL_PENDING_STATS.

See Also: "ALL_COL_PENDING_STATS" on page 2-29

DBA_COL_PRIVS

DBA_COL_PRIVS describes all column object grants in the database.

Related View

USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the column
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)

See Also: "USER_COL_PRIVS" on page 5-103

DBA_COLL_TYPES

DBA_COLL_TYPES describes all named collection types (arrays, nested tables, object tables, and so on) in the database. Its columns are the same as those in "ALL_COLL_TYPES" on page 2-31.

DBA_COMMON_AUDIT_TRAIL

DBA_COMMON_AUDIT_TRAIL displays all standard and fine-grained audit trail entries, mandatory and SYS audit records written in XML format.

Column	Datatype	NULL	Description
AUDIT_TYPE	VARCHAR2(18)		Audit trail type: ▪ Standard Audit ▪ Standard XML Audie ▪ Fine Grained Audit ▪ Fine Grained XML Audit ▪ SYS XML Audit ▪ Mandatory XML Audit
SESSION_ID	NUMBER		Numeric ID for the Oracle session
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
STATEMENTID	NUMBER		Numeric ID for the statement run; a statement may cause multiple audit records
ENTRYID	NUMBER		Numeric ID for the audit trail entry in the session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the audited operation (timestamp of user login for entries created by AUDIT SESSION) in the session's time zone
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
DB_USER	VARCHAR2(30)		Database user name of the user whose actions were audited
CLIENT_ID	VARCHAR2(64)		Client identifier in the Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
EXT_NAME	VARCHAR2(4000)		User external name
OS_USER	VARCHAR2(255)		Operating system login user name of the user whose actions were audited
USERHOST	VARCHAR2(128)		Client host machine name
OS_PROCESS	VARCHAR2(16)		Operating system process identifier of the Oracle process
TERMINAL	VARCHAR2(255)		Identifier of the user's terminal
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OBJECT_SCHEMA	VARCHAR2(30)		Owner of the audited object
OBJECT_NAME	VARCHAR2(128)		Name of the object affected by the action
POLICY_NAME	VARCHAR2(30)		Name of the Fine-Grained Auditing Policy
NEW_OWNER	VARCHAR2(30)		Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(128)		New name of the object after a RENAME or the name of the underlying object
ACTION	NUMBER		Numeric action type code. The corresponding name of the action type is in the STATEMENT_TYPE column.
STATEMENT_TYPE	VARCHAR2(28)		Name of the action type corresponding to the numeric code in the ACTION column
AUDIT_OPTION	VARCHAR2(40)		Auditing option set with the AUDIT statement
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object was accessed or modified
RETURNCODE	NUMBER		Oracle error code generated by the action (0 if the action succeeded)

Column	Datatype	NULL	Description
SCN	NUMBER		System change number (SCN) of the query
COMMENT_TEXT	VARCHAR2(4000)		Text comment on the audit trail entry, providing more information about the statement audited Also indicates how the user was authenticated: ■ DATABASE - Authentication was done by password ■ NETWORK - Authentication was done by Oracle Net Services or the Advanced Networking Option ■ PROXY - Client was authenticated by another user. The name of the proxy user follows the method type.
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
OBJ_PRIVILEGE	VARCHAR2(16)		Object privileges granted or revoked by a GRANT or REVOKE statement
SYS_PRIVILEGE	VARCHAR2(40)		System privileges granted or revoked by a GRANT or REVOKE statement
ADMIN_OPTION	VARCHAR2(1)		Indicates whether the role or system privilege was granted with the ADMIN option
OS_PRIVILEGE	VARCHAR2(7)		Operating System privilege used for Mandatory and SYS audit record
GRANTEE	VARCHAR2(30)		Name of the grantee specified in a GRANT or REVOKE statement
PRIV_USED	VARCHAR2(40)		System privilege used to execute the action
SES_ACTIONS	VARCHAR2(19)		Session summary (a string of 16 characters, one for each action type in the order ALTER, AUDIT, COMMENT, DELETE, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE, REFERENCES, and EXECUTE. Positions 14, 15, and 16 are reserved for future use. The characters are: ■ - - None ■ S - Success ■ F - Failure ■ B - Both
LOGOFF_TIME	DATE		Timestamp of user log off
LOGOFF_LREAD	NUMBER		Number of logical reads in the session
LOGOFF_PREAD	NUMBER		Number of physical reads in the session
LOGOFF_LWRITE	NUMBER		Number of logical writes for the session
LOGOFF_DLOCK	VARCHAR2(40)		Number of deadlocks detected during the session
SESSION_CPU	NUMBER		Amount of CPU time used by the Oracle session
OBJECT_EDITION	VARCHAR2(30)		???

DBA_COMPARISON

DBA_COMPARISON displays details about all comparison objects in the database.

Related View

USER_COMPARISON displays details about comparison objects owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of comparison

Column	Datatype	NULL	Description
COMPARISON_NAME	VARCHAR2(30)	NOT NULL	Name of comparison
COMPARISON_MODE	VARCHAR2(5)		Mode of comparison: TABLE
SCHEMA_NAME	VARCHAR2(30)		Schema name of local object
OBJECT_NAME	VARCHAR2(30)		Name of local object
OBJECT_TYPE	VARCHAR2(17)		Type of local object
REMOTE_SCHEMA_NAME	VARCHAR2(30)		Schema name of remote object
REMOTE_OBJECT_NAME	VARCHAR2(30)		Name of remote object
REMOTE_OBJECT_TYPE	VARCHAR2(17)		Type of remote object
DBLINK_NAME	VARCHAR2(128)		Database link name to remote database
SCAN_MODE	VARCHAR2(9)		Scan mode of comparison: FULL
SCAN_PERCENT	NUMBER		Scan percent of comparison: applicable to random and cyclic modes
CYCLIC_INDEX_VALUE	VARCHAR2(4000)		Last index column value used in a cyclic scan
NULL_VALUE	VARCHAR2(4000)		Value to use for null columns
LOCAL_CONVERGE_TAG	RAW(2000)		The local Streams tag used while performing converge DMLS
REMOTE_CONVERGE_TAG	RAW(2000)		The remote Streams tag used while performing converge DMLS
MAX_NUM_BUCKETS	NUMBER		Suggested maximum number of buckets in a scan
MIN_ROWS_IN_BUCKET	NUMBER		Suggested minimum number of rows in a bucket
LAST_UPATE_TIME	TIMESTAMP(6)		The time that this row was last updated

DBA_COMPARISON_COLUMNS

DBA_COMPARISON_COLUMNS displays details about the columns for all comparison objects in the database.

Related View

USER_COMPARISON_COLUMNS displays details about the columns for all comparison objects owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the comparison
COMPARISON_NAME	VARCHAR2(30)	NOT NULL	Name of the comparison
COLUMN_POSITION	NUMBER	NOT NULL	Position of the column
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Name of the column
INDEX_COLUMN	VARCHAR2(1)		Whether or not the column is an index column

DBA_COMPARISON_ROW_DIF

DBA_COMPARISON_ROW_DIF provides details about differing rows in all comparison scans in the database.

Related View

USER_COMPARISON_ROW_DIF provides details about differing rows in all comparison scans owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the comparison
COMPARISON_NAME	VARCHAR2(30)	NOT NULL	Name of the comparison
SCAN_ID	NUMBER	NOT NULL	Scan ID for the comparison scan
LOCAL_ROWID	ROWID		Local rowid of the differing row
REMOTE_ROWID	ROWID		Remote rowid of the differing row
INDEX_VALUE	VARCHAR2(4000)		Index column value of the differing row
STATUS	VARCHAR2(3)		Status of the differing row: SUC or DIF
LAST_UPDATE_TIME	TIMESTAMP(6)		The time that this row was last updated

DBA_COMPARISON_SCAN

DBA_COMPARISON_SCAN displays details about all comparison scans in the database.

Related View

USER_COMPARISON_SCAN displays details about all comparison scans owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of comparison scan
COMPARISON_NAME	VARCHAR2(30)	NOT NULL	Name of comparison scan
SCAN_ID	NUMBER	NOT NULL	Scan ID
PARENT_SCAN_ID	NUMBER		Scan ID of the immediate parent scan
ROOT_SCAN_ID	NUMBER		Scan ID of the root (top-most) parent
STATUS	VARCHAR2(16)		Status of scan: SUC, BUCKET DIF, FINAL BUCKET DIF, ROW DIF
CURRENT_DIF_COUNT	NUMBER		Current cumulative (including children) diff count of the scan
INITIAL_DIF_COUNT	NUMBER		Initial cumulative (including children) diff count of the scan
COUNT_ROWS	NUMBER		Number of rows in the scan
SCAN_NULLS	VARCHAR2(1)		Whether or not NULLs are part of this scan
LAST_UPDATE_TIME	TIMESTAMP(6)		The time that this row was last updated

DBA_COMPARISON_SCAN_VALUES

DBA_COMPARISON_SCAN_VALUES displays details about the values for all comparison scans in the database.

Related View

USER_COMPARISON_SCAN_VALUES displays details about the values for all comparison scans owned by the current user. This view does not have the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the comparison scan
COMPARISON_SCAN	VARCHAR2(30)	NOT NULL	Name of the comparison scan
SCAN_ID	NUMBER	NOT NULL	Scan ID

Column	Datatype	NULL	Description
COLUMN_POSITION	NUMBER	NOT NULL	Column position, as in DBA_COMPARISON_COLUMNS
MIN_VALUE	VARCHAR2(4000)		Minimum value of the scan
MAX_VALUE	VARCHAR2(4000)		Maximum value of the scan
LAST_UPDATE_TIME	TIMESTAMP(6)		The time that this row was last updated

DBA_CONNECT_ROLE_GRANTEES

DBA_CONNECT_ROLE_GRANTEES displays information about users who are granted the CONNECT privilege.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)		User or schema to which the CONNECT role is granted
PATH_OF_CONNECT_ROLE_GRANT	VARCHAR2(4000)		The path of role inheritance through which the grantee is granted the CONNECT role
ADMIN_OPT	VARCHAR2(3)		Whether or not the grantee was granted the ADMIN option for the CONNECT role

DBA_CONS_OBJ_COLUMNS

DBA_CONS_OBJ_COLUMNS displays information about the types that object columns (or attributes) or collection elements have been constrained to, in all tables in the database. Its columns are the same as those in ALL_CONS_OBJ_COLUMNS.

See Also: "ALL_CONS_OBJ_COLUMNS" on page 2-32

DBA_CONSTRAINTS

DBA_CONSTRAINTS describes all constraint definitions on all tables in the database. Its columns are the same as those in "ALL_CONSTRAINTS" on page 2-32.

DBA_CONTEXT

DBA_CONTEXT provides all context namespace information in the database. Its columns are the same as those in "ALL_CONTEXT" on page 2-34.

DBA_CPOOL_INFO

DBA_CPOOL_INFO displays configuration information about all Database Resident Connection Pools in the database.

Column	Datatype	NULL	Description
CONNECTION_POOL	VARCHAR2(128)		Name of the connection pool
STATUS	VARCHAR2(16)		Status of the pool: ▪ ACTIVE ▪ INACTIVE
MINSIZE	NUMBER		Minimum number of pooled servers that are always alive in the pool
MAXSIZE	NUMBER		Maximum number of pooled servers in the pool

Column	Datatype	NULL	Description
INCRSIZE	NUMBER		Number of pooled servers by which the pool is incremented if servers are unavailable when a client application request is received
SESSION_CACHED_CURSORS	NUMBER		Number of session cursors to cache in each pooled server
INACTIVITY_TIMEOUT	NUMBER		Maximum time (in seconds) that the pooled server can stay idle in the pool. After this time, the server is terminated.
MAX_THINK_TIME	NUMBER		Maximum time of inactivity (in seconds) for a client after obtaining a pooled server from the pool. After obtaining a pooled server from the pool, if the client application does not issue a database call for the time specified by this column, then the pooled server is freed and the client connection is terminated.
MAX_USE_SESSION	NUMBER		Number of times a pooled server can be taken and released to the pool
MAX_LIFETIME_SESSION	NUMBER		Time (in seconds) for a pooled server to live in the pool

DBA_CPU_USAGE_STATISTICS

DBA_CPU_USAGE_STATISTICS displays database CPU usage statistics.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
VERSION	VARCHAR2(17)	NOT NULL	Database version
TIMESTAMP	DATE	NOT NULL	Time at which the CPU usage changed
CPU_COUNT	NUMBER		CPU count of the database
CPU_CORE_COUNT	NUMBER		CPU core count of the database
CPU_SOCKET_COUNT	NUMBER		CPU socket count of the database

DBA_CQ_NOTIFICATION_QUERIES

DBA_CQ_NOTIFICATION_QUERIES describes the registered queries for all CQ notifications in the database.

Related View

USER_CQ_NOTIFICATION_QUERIES describes the registered queries for the CQ notifications owned by the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
QUERYID	NUMBER		ID of the query
QUERYTEXT	CLOB		Text of the query
REGID	NUMBER		Registration ID that the query is registered with
USERNAME	VARCHAR2(31)		Name of the user who registered the query

See Also: "USER_CQ_NOTIFICATION_QUERIES" on page 5-105

DBA_CUBE_ATTR_VISIBILITY

DBA_CUBE_ATTR_VISIBILITY describes all OLAP attributes visible for the dimensions, hierarchies, and levels in the database. Its columns are the same as those in ALL_CUBE_ATTR_VISIBILITY.

See Also: "ALL_CUBE_ATTR_VISIBILITY" on page 2-34

DBA_CUBE_ATTRIBUTES

DBA_CUBE_ATTRIBUTES describes the attributes for all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_ATTRIBUTES.

See Also: "ALL_CUBE_ATTRIBUTES" on page 2-34

DBA_CUBE_BUILD_PROCESSES

DBA_CUBE_BUILD_PROCESSES describes all OLAP build processes and maintenance scripts in the database. Its columns are the same as those in ALL_CUBE_BUILD_PROCESSES.

See Also: "ALL_CUBE_BUILD_PROCESSES" on page 2-35

DBA_CUBE_CALCULATED_MEMBERS

DBA_CUBE_CALCULATED_MEMBERS describes the calculated members for all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_CALCULATED_MEMBERS.

See Also: "ALL_CUBE_CALCULATED_MEMBERS" on page 2-36

DBA_CUBE_DIM_LEVELS

DBA_CUBE_DIM_LEVELS describes all OLAP dimension levels in the database. Its columns are the same as those in ALL_CUBE_DIM_LEVELS.

See Also: "ALL_CUBE_DIM_LEVELS" on page 2-36

DBA_CUBE_DIM_MODELS

DBA_CUBE_DIM_MODELS describes the models for all OLAP dimensions in the database. Its columns are the same as those in ALL_CUBE_DIM_MODELS.

See Also: "ALL_CUBE_DIM_MODELS" on page 2-37

DBA_CUBE_DIM_VIEW_COLUMNS

DBA_CUBE_DIM_VIEW_COLUMNS describes the columns of the relational views of all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_DIM_VIEW_COLUMNS.

See Also: "ALL_CUBE_DIM_VIEW_COLUMNS" on page 2-37

DBA_CUBE_DIM_VIEWS

DBA_CUBE_DIM_VIEWS describes the relational views of all OLAP dimensions in the database. Its columns are the same as those in ALL_CUBE_DIM_VIEWS.

See Also: "ALL_CUBE_DIM_VIEWS" on page 2-38

DBA_CUBE_DIMENSIONALITY

DBA_CUBE_DIMENSIONALITY describes the dimension order for all OLAP cubes in the database. Its columns are the same as those in ALL_CUBE_DIMENSIONALITY.

See Also: "ALL_CUBE_DIMENSIONALITY" on page 2-38

DBA_CUBE_DIMENSIONS

DBA_CUBE_DIMENSIONS describes all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_DIMENSIONS.

See Also: "ALL_CUBE_DIMENSIONS" on page 2-39

DBA_CUBE_HIER_LEVELS

DBA_CUBE_HIER_LEVELS describes the hierarchy levels for all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_HIER_LEVELS.

See Also: "ALL_CUBE_HIER_LEVELS" on page 2-39

DBA_CUBE_HIER_VIEW_COLUMNS

DBA_CUBE_HIER_VIEW_COLUMNS describes the columns of the relational hierarchy views of all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_HIER_VIEW_COLUMNS.

See Also: "ALL_CUBE_HIER_VIEW_COLUMNS" on page 2-40

DBA_CUBE_HIER_VIEWS

DBA_CUBE_HIER_VIEWS describes the hierarchies for all OLAP cube dimensions in the database. Its columns are the same as those in ALL_CUBE_HIER_VIEWS.

See Also: "ALL_CUBE_HIER_VIEWS" on page 2-41

DBA_CUBE_HIERARCHIES

DBA_CUBE_HIERARCHIES describes all OLAP dimension hierarchies in the database. Its columns are the same as those in ALL_CUBE_HIERARCHIES.

See Also: "ALL_CUBE_HIERARCHIES" on page 2-41

DBA_CUBE_MEASURES

DBA_CUBE_MEASURES describes the measures for all OLAP cubes in the database. Its columns are the same as those in ALL_CUBE_MEASURES.

See Also: "ALL_CUBE_MEASURES" on page 2-42

DBA_CUBE_VIEW_COLUMNS

DBA_CUBE_VIEW_COLUMNS describes the columns of relational views of all OLAP cubes in the database. Its columns are the same as those in ALL_CUBE_VIEW_COLUMNS.

See Also: "ALL_CUBE_VIEW_COLUMNS" on page 2-43

DBA_CUBE_VIEWS

DBA_CUBE_VIEWS describes the relational views of all OLAP cubes in the database. Its columns are the same as those in ALL_CUBE_VIEWS.

See Also: "ALL_CUBE_VIEWS" on page 2-43

DBA_CUBES

DBA_CUBES describes all OLAP cubes in the database. Its columns are the same as those in ALL_CUBES.

See Also: "ALL_CUBES" on page 2-44

DBA_DATA_FILES

DBA_DATA_FILES describes database files.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database file
FILE_ID	NUMBER	NOT NULL	File identifier number of the database file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file in bytes
BLOCKS	NUMBER	NOT NULL	Size of the file in Oracle blocks
STATUS	VARCHAR2(9)		File status: AVAILABLE or INVALID (INVALID means that the file number is not in use, for example, a file in a tablespace that was dropped)
RELATIVE_FNO	NUMBER		Relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Autoextensible indicator
MAXBYTES	NUMBER		Maximum file size in bytes
MAXBLOCKS	NUMBER		Maximum file size in blocks
INCREMENT_BY	NUMBER		Number of tablespace blocks used as autoextension increment. Block size is contained in the BLOCK_SIZE column of the DBA_TABLESPACES view.
USER_BYTES	NUMBER		The size of the file available for user data. The actual size of the file minus the USER_BYTES value is used to store file related metadata.
USER_BLOCKS	NUMBER		Number of blocks which can be used by the data

Column	Datatype	NULL	Description
ONLINE_STATUS	VARCHAR2(7)		Online status of the file: ▪ SYSOFF ▪ SYSTEM ▪ OFFLINE ▪ ONLINE ▪ RECOVER

DBA_DATAPUMP_JOBS

DBA_DATAPUMP_JOBS identifies all active Data Pump jobs in the database, regardless of their state, on an instance (or on all instances for Real Application Clusters). It also show all Data Pump master tables not currently associated with an active job.

Related View

USER_DATAPUMP_JOBS displays the Data Pump jobs owned by the current user. This view does not display the OWNER_NAME column.

Column	Datatype	NULL	Description
OWNER_NAME	VARCHAR2(30)		User that initiated the job
JOB_NAME	VARCHAR2(30)		User-supplied name for the job (or the default name generated by the server)
OPERATION	VARCHAR2(30)		Type of job
JOB_MODE	VARCHAR2(30)		Mode of job
STATE	VARCHAR2(30)		Current job state
DEGREE	NUMBER		Number of worker processes performing the operation
ATTACHED_SESSIONS	NUMBER		Number of sessions attached to the job
DATAPUMP_SESSIONS	NUMBER		Number of Data Pump sessions participating in the job

See Also: "USER_DATAPUMP_JOBS" on page 5-107

DBA_DATAPUMP_SESSIONS

DBA_DATAPUMP_SESSIONS identifies the user sessions that are attached to a Data Pump job. The information in this view is useful for determining why a stopped Data Pump operation has not gone away.

Column	Datatype	NULL	Description
OWNER_NAME	VARCHAR2(30)		User that initiated the job
JOB_NAME	VARCHAR2(30)		User-supplied name for the job (or the default name generated by the server)
SADDR	RAW(4 | 8)		Address of the session attached to the job. Can be used with V$SESSION view.

Column	Datatype	NULL	Description
SESSION_TYPE	VARCHAR2(14)		Data Pump session type: ■ DBMS_DATAPUMP - Data Pump interface process (one for each active instantiation of DBMS_DATAPUMP.OPEN and DBMS_DATAPUMP.ATTACH per job.) ■ MASTER - master control process (one per job) ■ WORKER - worker process (1 to *n* per job, depending on degree of parallelism) ■ EXTERNAL TABLE - external table data access process (1 to *n*, depending on degree of parallelism, for jobs that use external tables as the data access method for some tables) ■ OTHER

DBA_DB_LINKS

DBA_DB_LINKS describes all database links in the database. Its columns (except for PASSWORD) are the same as those in ALL_DB_LINKS.

See Also: "ALL_DB_LINKS" on page 2-44

DBA_DDL_LOCKS

DBA_DDL_LOCKS lists all DDL locks held in the database and all outstanding requests for a DDL lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session identifier
OWNER	VARCHAR2(30)		Owner of the lock
NAME	VARCHAR2(30)		Name of the lock
TYPE	VARCHAR2(40)		Lock type: ■ Cursor ■ Table/Procedure/Type ■ Body ■ Trigger ■ Index ■ Cluster ■ Java Source ■ Java Resource ■ Java Data
MODE_HELD	VARCHAR2(9)		Lock mode: ■ None ■ Null ■ Share ■ Exclusive
MODE_REQUESTED	VARCHAR2(9)		Lock request type: ■ None ■ Null ■ Share ■ Exclusive

DBA_DEPENDENCIES

DBA_DEPENDENCIES describes all dependencies in the database between procedures, packages, functions, package bodies, and triggers, including dependencies on views created without any database links. Its columns are the same as those in "ALL_DEPENDENCIES" on page 2-46.

DBA_DIM_ATTRIBUTES

DBA_DIM_ATTRIBUTES describes the relationships between dimension levels and functionally dependent columns in the database. The level columns and the dependent column must be in the same table. This view's columns are the same as those in "ALL_DIM_ATTRIBUTES" on page 2-47.

DBA_DIM_CHILD_OF

DBA_DIM_CHILD_OF describes a hierarchical relationship of 1 to *n* between all the pairs of levels in the dimensions in the database. Its columns are the same as those in "ALL_DIM_CHILD_OF" on page 2-47.

DBA_DIM_HIERARCHIES

DBA_DIM_HIERARCHIES describes all the dimension hierarchies in the database. Its columns are the same as those in "ALL_DIM_HIERARCHIES" on page 2-48.

DBA_DIM_JOIN_KEY

DBA_DIM_JOIN_KEY describes all joins in the database between two dimension tables. The join is always specified between a parent dimension level column and a child column. This view's columns are the same as those in "ALL_DIM_JOIN_KEY" on page 2-48.

DBA_DIM_LEVEL_KEY

DBA_DIM_LEVEL_KEY describes the columns of all dimension levels in the database. This view's columns are the same as those in "ALL_DIM_LEVEL_KEY" on page 2-49.

DBA_DIM_LEVELS

DBA_DIM_LEVELS describes all dimension levels in the database. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "ALL_DIM_LEVELS" on page 2-49.

DBA_DIMENSIONS

DBA_DIMENSIONS represents dimension objects. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-49.

DBA_DIRECTORIES

DBA_DIRECTORIES describes all directory objects in the database. Its columns are the same as those in "ALL_DIRECTORIES" on page 2-50.

DBA_DML_LOCKS

DBA_DML_LOCKS lists all DML locks held in the database and all outstanding requests for a DML lock.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
OWNER	VARCHAR2(30)	NOT NULL	Owner of the lock
NAME	VARCHAR2(30)	NOT NULL	Name of the lock
MODE_HELD	VARCHAR2(13)		The type of lock held. The values are: ■ ROWS_S (SS): row share lock ■ ROW-X (SX): row exclusive lock ■ SHARE (S): share lock ■ S/ROW-X (SSX): exclusive lock ■ NONE: lock requested but not yet obtained
MODE_REQUESTED	VARCHAR2(13)		Lock request type. The values are: ■ ROWS_S (SS): row share lock ■ ROW-X (SX): row exclusive lock ■ SHARE (S): share lock ■ S/ROW-X (SSX): exclusive lock ■ NONE: Lock identifier obtained; lock not held or requested
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Blocking others

DBA_DMT_FREE_SPACE

DBA_DMT_FREE_SPACE describes the free extents in all dictionary managed tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_ID	NUMBER	NOT NULL	Identifier number of the tablespace containing the extent
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK_ID	NUMBER	NOT NULL	Starting block number of the extent
BLOCKS	NUMBER	NOT NULL	Size of the extent (in Oracle blocks)

DBA_DMT_USED_EXTENTS

DBA_DMT_USED_EXTENTS describes the extents comprising the segments in all dictionary managed tablespaces in the database.

Column	Datatype	NULL	Description
SEGMENT_FILEID	NUMBER	NOT NULL	File number of the segment header of the extent
SEGMENT_BLOCK	NUMBER	NOT NULL	Block number of the segment header of the extent
TABLESPACE_ID	NUMBER	NOT NULL	Identifier number of the tablespace containing the extent
EXTENT_ID	NUMBER	NOT NULL	Extent number in the segment
FILEID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK	NUMBER	NOT NULL	Starting block number of the extent
LENGTH	NUMBER	NOT NULL	Number of blocks in the extent

DBA_ENABLED_AGGREGATIONS

DBA_ENABLED_AGGREGATIONS displays information about enabled on-demand statistic aggregation.

Column	Datatype	NULL	Description
AGGREGATION_TYPE	VARCHAR2(21)		Type of the aggregation: ■ CLIENT_ID ■ SERVICE ■ SERVICE_MODULE ■ SERVICE_MODULE_ACTION
PRIMARY_ID	VARCHAR2(64)		Primary qualifier (specific client identifier or service name)
QUALIFIER_ID1	VARCHAR2(48)		Secondary qualifier (specific module name)
QUALIFIER_ID2	VARCHAR2(32)		Additional qualifier (specific action name)

DBA_ENABLED_TRACES

DBA_ENABLED_TRACES displays information about enabled SQL traces.

Column	Datatype	NULL	Description
TRACE_TYPE	VARCHAR2(21)		Type of the trace: ■ CLIENT_ID ■ SERVICE ■ SERVICE_MODULE ■ SERVICE_MODULE_ACTION ■ DATABASE
PRIMARY_ID	VARCHAR2(64)		Primary qualifier (specific client identifier or service name)
QUALIFIER_ID1	VARCHAR2(48)		Secondary qualifier (specific module name)
QUALIFIER_ID2	VARCHAR2(32)		Additional qualifier (specific action name)
WAITS	VARCHAR2(5)		Indicates whether waits are traced (TRUE) or not (FALSE)
BINDS	VARCHAR2(5)		Indicates whether binds are traced (TRUE) or not (FALSE)
INSTANCE_NAME	VARCHAR2(16)		Instance name for tracing restricted to named instances

DBA_ENCRYPTED_COLUMNS

DBA_ENCRYPTED_COLUMNS maintains encryption algorithm information for all encrypted columns in the database. Its columns are the same as those in "ALL_ENCRYPTED_COLUMNS" on page 2-50.

DBA_EPG_DAD_AUTHORIZATION

DBA_EPG_DAD_AUTHORIZATION describes the DADs that are authorized to use different user's privileges.

Related View

USER_EPG_DAD_AUTHORIZATION describes the DADs that are authorized to use the user's privileges. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
DAD_NAME	VARCHAR2(64)	NOT NULL	Name of DAD
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user whose privileges the DAD is authorized to use

DBA_ERRORS

DBA_ERRORS describes the current errors on all stored objects in the database. Its columns are the same as those in ALL_ERRORS.

See Also: "ALL_ERRORS" on page 2-51

DBA_EVALUATION_CONTEXT_TABLES

DBA_EVALUATION_CONTEXT_TABLES describes the tables in all rule evaluation contexts in the database. Its columns are the same as those in ALL_EVALUATION_CONTEXT_TABLES.

See Also: "ALL_EVALUATION_CONTEXT_TABLES" on page 2-51

DBA_EVALUATION_CONTEXT_VARS

DBA_EVALUATION_CONTEXT_VARS describes the variables in all rule evaluation contexts in the database. Its columns are the same as those in ALL_EVALUATION_CONTEXT_VARS.

See Also: "ALL_EVALUATION_CONTEXT_VARS" on page 2-52

DBA_EVALUATION_CONTEXTS

DBA_EVALUATION_CONTEXTS describes all rule evaluation contexts in the database. Its columns are the same as those in ALL_EVALUATION_CONTEXTS.

See Also: "ALL_EVALUATION_CONTEXTS" on page 2-52

DBA_EXP_FILES

DBA_EXP_FILES describes export files.

Column	Datatype	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the export session
EXP_TYPE	VARCHAR2(11)		Type of export file: complete, cumulative, or incremental
FILE_NAME	VARCHAR2(100)	NOT NULL	Name of the export file
USER_NAME	VARCHAR2(30)	NOT NULL	Name of user who executed export
TIMESTAMP	DATE	NOT NULL	Timestamp of the export session

DBA_EXP_OBJECTS

DBA_EXP_OBJECTS describes objects that have been incrementally exported.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of exported object

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of exported object
OBJECT_TYPE	VARCHAR2(12)		Type of exported object
CUMULATIVE	DATE		Timestamp of last cumulative export
INCREMENTAL	DATE	NOT NULL	Timestamp of last incremental export
EXPORT_VERSION	NUMBER(3)	NOT NULL	The ID of the export session

DBA_EXP_VERSION

DBA_EXP_VERSION displays the version number of the last export session.

Column	Datatype	NULL	Description
EXP_VERSION	NUMBER(3)	NOT NULL	Version number of the last export session

DBA_EXTENTS

DBA_EXTENTS describes the extents comprising the segments in all tablespaces in the database.

Note that if a datafile (or entire tablespace) is offline in a locally managed tablespace, you will not see any extent information. If an object has extents in an online file of the tablespace, you will see extent information about the offline datafile. However, if the object is entirely in the offline file, a query of this view will not return any records.

Related View

USER_EXTENTS describes the extents comprising the segments owned by the current user's objects. This view does not display the OWNER, FILE_ID, BLOCK_ID, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the segment associated with the extent
SEGMENT_NAME	VARCHAR2(81)		Name of the segment associated with the extent
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)
SEGMENT_TYPE	VARCHAR2(18)		Type of the segment: INDEX PARTITION, TABLE PARTITION
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent
EXTENT_ID	NUMBER		Extent number in the segment
FILE_ID	NUMBER		File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Starting block number of the extent
BYTES	NUMBER		Size of the extent in bytes
BLOCKS	NUMBER		Size of the extent in Oracle blocks
RELATIVE_FNO	NUMBER		Relative file number of the first extent block

See Also: "USER_EXTENTS" on page 5-109

DBA_EXTERNAL_LOCATIONS

DBA_EXTERNAL_LOCATIONS describes the locations (data sources) of all external tables in the database. Its columns are the same as those in ALL_EXTERNAL_LOCATIONS.

See Also: "ALL_EXTERNAL_LOCATIONS" on page 2-53

DBA_EXTERNAL_TABLES

DBA_EXTERNAL_TABLES describes all external tables in the database. Its columns are the same as those in ALL_EXTERNAL_TABLES.

See Also: "ALL_EXTERNAL_TABLES" on page 2-53

DBA_FEATURE_USAGE_STATISTICS

DBA_FEATURE_USAGE_STATISTICS displays information about database feature usage statistics.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database identifier of the database being tracked
NAME	VARCHAR2(64)	NOT NULL	Name of the feature (see Table 4–1)
VERSION	VARCHAR2(17)	NOT NULL	Database version in which the feature was tracked
DETECTED_USAGES	NUMBER	NOT NULL	Number of times the system has detected usage for the feature
TOTAL_SAMPLES	NUMBER	NOT NULL	Number of times the system has woken up and checked for feature usage
CURRENTLY_USED	VARCHAR2(5)		Indicates whether usage was detected the last time the system checked (TRUE) or not (FALSE)
FIRST_USAGE_DATE	DATE		First sample time the system detected usage of the feature
LAST_USAGE_DATE	DATE		Last sample time the system detected usage of the feature
AUX_COUNT	NUMBER		This column stores feature-specific usage data in number format.
FEATURE_INFO	CLOB		This column stores feature-specific usage data in character format.
LAST_SAMPLE_DATE	DATE		Amount of time (in seconds) between the last two usage sample times
LAST_SAMPLE_PERIOD	NUMBER		Amount of time (in hours) between the last two usage sample times
SAMPLE_INTERVAL	NUMBER		Sample interval
DESCRIPTION	VARCHAR2(128)		Description of the feature and usage detection logic (see Table 4–1)

Table 4–1 DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Undo Advisor	Undo Advisor has been used.
SQL Tuning Advisor	SQL Tuning Advisor has been used.
Segment Advisor	Segment Advisor has been used.
SQL Workload Manager	SQL Workload Manager has been used.

Table 4–1 (Cont.) DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Tune MView	Tune MView has been used.
SQL Performance Impact	SQL Performance Impact has been used.
SQL Repair Advisor	SQL Repair Advisor has been used.
SQL Tuning Set (user)	A SQL Tuning Set has been created in the database in a user schema.
SQL Tuning Set (system)	A SQL Tuning Set has been created in the database in the SYS schema.
Automatic SQL Tuning Advisor	Automatic SQL Tuning Advisor has been used.
SQL Profile	SQL profiles have been used.
Database Replay: Workload Capture	Database Replay: Workload was ever captured.
Database Replay: Workload Replay	Database Replay: Workload was ever replayed.
Streams (system)	Oracle Streams processes have been configured
Streams (user)	Users have configured Oracle Streams AQ
Transparent Gateway	Heterogeneous Connectivity, access to a non-Oracle system, has been configured.
Virtual Private Database (VPD)	Virtual Private Database (VPD) policies are being used.
Workspace Manager	There is at least one version enabled table.
XDB	XDB feature is being used.
Application Express	Application Express feature is being used.
LOB	Persistent LOBs are being used.
Object	Object feature is being used.
Extensibility	Extensibility feature is being used.
SQL Plan Management	SQL Plan Management has been used.
Rules Manager	Rules Manager and Expression Filter
Oracle Utility Datapump (Export)	Oracle Utility Datapump (Export) has been used.
Oracle Utility Datapump (Import)	Oracle Utility Datapump (Import) has been used.
Result Cache	The Result Cache feature has been used.
Oracle Utility SQL Loader (Direct Path Load)	Oracle Utility SQL Loader (Direct Path Load) has been used.
Oracle Utility Metadata API	Oracle Utility (Metadata API) has been used.
Transparent Data Encryption	Transparent Database Encryption is being used. There is atleast one column or tablespace that is encrypted.
interMedia	interMedia has been used
interMedia DICOM	interMedia DICOM has been used
Materialized Views (User)	User Materialized Views exist in the database
Change Data Capture	Change Data Capture exit in the database
Services	Oracle Services.
Semantics/RDF	A semantic network has been created indicating usage of the Oracle Semantics Feature.
SecureFiles	SecureFiles are being used
Automatic Segment Advisor	Automatic Segment Advisor has been used.
Segment Shrink	Segment Shrink has been used.
Advanced Replication	Advanced Replication has been enabled.
Advanced Security	External Global users are configured.
Audit Options	Audit options in use.
Automatic Maintenance - Optimizer Statistics Gathering	Automatic initiation of Optimizer Statistics Collection

Table 4–1 (Cont.) DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Automatic Maintenance - Space Advisor	Automatic initiation of Space Advisor
Automatic Maintenance - SQL Tuning Advisor	Automatic initiation of SQL Tuning Advisor
Automatic Segment Space Management (system)	Extents of locally managed tablespaces are managed automatically by Oracle.
Automatic Segment Space Management (user)	Extents of locally managed user tablespaces are managed automatically by Oracle.
Automatic SQL Execution Memory	Sizing of work areas for all dedicated sessions (PGA) is automatic.
Automatic Storage Manager	Automatic Storage Management has been enabled
Automatic Undo Management	Oracle automatically manages undo data using an UNDO tablespace.
Automatic Workload Repository	A manual Automatic Workload Repository (AWR) snapshot was taken in the last sample period.
AWR Baseline	At least one AWR Baseline has been created by the user
AWR Baseline Template	At least one AWR Baseline Template has been created by the user
Backup Encryption	Encrypted backups are being used.
Baseline Adaptive Thresholds	Adaptive Thresholds have been configured.
Baseline Static Computations	Static baseline statistics have been computed.
Change-Aware Incremental Backup	Track blocks that have changed in the database.
Client Identifier	Application User Proxy Authentication: Client Identifier is used at this specific time.
Clusterwide Global Transactions	Clusterwide Global Transactions is being used.
CSSCAN	Oracle Database has been scanned at least once for character set:CSSCAN has been run at least once.
Character Semantics	Character length semantics is used in Oracle Database
Character Set	Character set is used in Oracle Database
Data Guard	Data Guard, a set of services, is being used to create, maintain, manage, and monitor one or more standby databases.
Data Mining	There exist Oracle Data Mining models in the database.
Dynamic SGA	The Oracle SGA has been dynamically resized through an ALTER SYSTEM SET statement.
EM Database Control	EM Database Control Home Page has been visited at least once.
EM Grid Control	EM Grid Control Database Home Page has been visited at least once.
File Mapping	File Mapping, the mechanism that shows a complete mapping of a file to logical volumes and physical devices, is being used.
Flashback Database	Flashback Database, a rewind button for the database, is enabled
Flashback Data Archive	Flashback Data Archive, a historical repository of changes to data contained in a table, is used
Internode Parallel Execution	Internode Parallel Execution is being used.
Label Security	Oracle Label Security, that enables label-based access control Oracle applications, is being used.
Locally Managed Tablespaces (system)	There exists tablespaces that are locally managed in the database.
Locally Managed Tablespaces (user)	There exists user tablespaces that are locally managed in the database.
Messaging Gateway	Messaging Gateway, that enables communication between non-Oracle messaging systems and Advanced Queuing (AQ), link configured.
Very Large Memory	Very Large Memory is enabled.
Automatic Memory Tuning	Automatic Memory Tuning is enabled.

Table 4–1 (Cont.) DBA_FEATURE_USAGE_STATISTICS Statistics

Name	Description
Automatic SGA Tuning	Automatic SGA Tuning is enabled.
Encrypted Tablespaces	Encrypted Tablespaces is enabled.
MTTR Advisor	Mean Time to Recover Advisor is enabled.
Multiple Block Sizes	Multiple Block Sizes are being used with this database.
OLAP - Analytic Workspaces	OLAP - the analytic workspaces stored in the database.
OLAP - Cubes	OLAP - number of cubes in the OLAP catalog that are fully mapped and accessible by the OLAP API.
Oracle Managed Files	Database files are being managed by Oracle.
Oracle Secure Backup	Oracle Secure Backup is used for backups to tertiary storage.
Parallel SQL DDL Execution	Parallel SQL DDL Execution is being used.
Parallel SQL DML Execution	Parallel SQL DML Execution is being used.
Parallel SQL Query Execution	Parallel SQL Query Execution is being used.
Partitioning (system)	Oracle Partitioning option is being used - there is at least one partitioned object created.
Partitioning (user)	Oracle Partitioning option is being used - there is at least one user partitioned object created.
Oracle Text	Oracle Text is being used - there is at least one oracle text index
PL/SQL Native Compilation	PL/SQL Native Compilation is being used - there is at least one natively compiled PL/SQL library unit in the database.
Real Application Clusters (RAC)	Real Application Clusters (RAC) is configured.
Recovery Area	The recovery area is configured.
Recovery Manager (RMAN)	Recovery Manager (RMAN) is being used to backup the database.
RMAN - Disk Backup	Recovery Manager (RMAN) is being used to backup the database to disk.
RMAN - Tape Backup	Recovery Manager (RMAN) is being used to backup the database to tape.
Block Media Recovery	Block Media Recovery is being used to repair the database.
Restore Point	Restore Points are being used as targets for Flashback
Logfile Multiplexing	Multiple members are used in a single log file group
Bigfile Tablespace	Bigfile tablespace is being used
Transportable Tablespace	Transportable tablespace is being used
Read Only Tablespace	Read only tablespace is being used
Deferred Open Read Only	Deferred open read only feature is enabled
Physical Standby with Real-Time Query	Physical standby database is in real-time query mode
Backup Rollforward	Backup Rollforward strategy is being used to backup the database.
Data Recovery Advisor	Data Recovery Advisor (DRA) is being used to repair the database.
Resource Manager	Oracle Database Resource Manager is being used to manage database resources.
Server Parameter File	The server parameter file (SPFILE) was used to startup the database.
Shared Server	The database is configured as Shared Server, where one server process can service multiple client programs.
Spatial	There is at least one usage of the Oracle Spatial index metadata table.
ADDM	ADDM has been used.
SQL Access Advisor	SQL Access Advisor has been used.

DBA_FGA_AUDIT_TRAIL

`DBA_FGA_AUDIT_TRAIL` displays all audit records for fine-grained auditing.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER	NOT NULL	Session id of the query
TIMESTAMP	DATE		Date and time of the query in the local database session time zone
DB_USER	VARCHAR2(30)		Database username who executed the query
OS_USER	VARCHAR2(255)		Operating system username who executed the query
USERHOST	VARCHAR2(128)		Client host machine name
CLIENT_ID	VARCHAR2(64)		Client identifier in each Oracle session
ECONTEXT_ID	VARCHAR2(64)		Application execution context identifier
EXT_NAME	VARCHAR2(4000)		External name
OBJECT_SCHEMA	VARCHAR2(30)		Owner of the table or view
OBJECT_NAME	VARCHAR2(128)		Name of the table or view
POLICY_NAME	VARCHAR2(30)		Name of the Fine-Grained Auditing Policy
SCN	NUMBER		System change number (SCN) of the query
SQL_TEXT	NVARCHAR2(2000)		SQL text of the query
SQL_BIND	NVARCHAR2(2000)		Bind variable data of the query
COMMENT$TEXT	VARCHAR2(4000)		Comments
STATEMENT_TYPE	VARCHAR2(7)		Statement type of the query: ■ SELECT ■ INSERT ■ UPDATE ■ DELETE
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Timestamp of the query in UTC (Coordinated Universal Time) time zone
PROXY_SESSIONID	NUMBER		Proxy session serial number, if an enterprise user has logged in through the proxy mechanism
GLOBAL_UID	VARCHAR2(32)		Global user identifier for the user, if the user has logged in as an enterprise user
INSTANCE_NUMBER	NUMBER		Instance number as specified by the INSTANCE_NUMBER initialization parameter
OS_PROCESS	VARCHAR2(16)		Operating System process identifier of the Oracle process
TRANSACTIONID	RAW(8)		Transaction identifier of the transaction in which the object is accessed or modified
STATEMENTID	NUMBER		Numeric ID for each statement run (a statement may cause many actions)
ENTRYID	NUMBER		Numeric ID for each audit trail entry in the session
OBJECT_EDITION	VARCHAR2(30)		???

Note: The SQL_BIND and SQL_TEXT columns are populated only if the policy has been created with the AUDIT_TRAIL parameter set to DB_EXTENDED.

DBA_FILE_GROUP_EXPORT_INFO

DBA_FILE_GROUP_EXPORT_INFO shows export-related information for each version in the database that has a valid Data Pump dump file. Its columns are the same as those in "ALL_FILE_GROUP_EXPORT_INFO" on page 2-54.

DBA_FILE_GROUP_FILES

DBA_FILE_GROUP_FILES shows the file set for each versioned file group in the database. Its columns are the same as those in "ALL_FILE_GROUP_FILES" on page 2-55.

DBA_FILE_GROUP_TABLES

DBA_FILE_GROUP_TABLES shows information about all the tables in the database that can be imported using the file set. Its columns are the same as those in "ALL_FILE_GROUP_TABLES" on page 2-55.

DBA_FILE_GROUP_TABLESPACES

DBA_FILE_GROUP_TABLESPACES shows information about the transportable tablespaces present (partially or completely) in all file sets in the database (when the file set contains dump files). Its columns are the same as those in"ALL_FILE_GROUP_TABLESPACES" on page 2-56.

DBA_FILE_GROUP_VERSIONS

DBA_FILE_GROUP_VERSIONS shows top-level version information for all file groups in the database. Its columns are the same as those in "ALL_FILE_GROUP_VERSIONS" on page 2-56.

DBA_FILE_GROUPS

DBA_FILE_GROUPS shows top-level metadata about all file groups in the database. Its columns are the same as those for "ALL_FILE_GROUPS" on page 2-57.

DBA_FLASHBACK_ARCHIVE

DBA_FLASHBACK_ARCHIVE describes all flashback archives available in the database.

Related View

USER_FLASHBACK_ARCHIVE describes the flashback archives available to the current user.

Column	Datatype	NULL	Description
FLASHBACK_ARCHIVE_NAME	VARCHAR2(255)	NOT NULL	Name of the flashback archive
FLASHBACK_ARCHIVE#	NUMBER	NOT NULL	Number of the flashback archive
RETENTION_IN_DAYS	NUMBER	NOT NULL	Maximum duration (in days) for which data is retained in the flashback archive
CREATE_TIME	TIMESTAMP(9)		Time at which the flashback archive was created
LAST_PURGE_TIME	TIMESTAMP(9)		Time at which the data in the flashback archive was last purged by the system
STATUS	VARCHAR2(7)		Indicates whether the flashback archive is a default flashback archive for the system (DEFAULT) or not (NULL)

See Also: "USER_FLASHBACK_ARCHIVE" on page 5-110

DBA_FLASHBACK_ARCHIVE_TABLES

DBA_FLASHBACK_ARCHIVE_TABLES displays information about all tables in the database that are enabled for Flashback Archive.

Related View

USER_FLASHBACK_ARCHIVE_TABLES displays information about the tables owned by the current user that are enabled for Flashback Archive.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table enabled for Flashback Archive
OWNER_NAME	VARCHAR2(30)	NOT NULL	Owner name of the table enabled for Flashback Archive
FLASHBACK_ARCHIVE_NAME	VARCHAR2(255)	NOT NULL	Name of the flashback archive
ARCHIVE_TABLE_NAME	VARCHAR2(53)		Name of the archive table containing the historical data for the user table

See Also: "USER_FLASHBACK_ARCHIVE_TABLES" on page 5-110

DBA_FLASHBACK_ARCHIVE_TS

DBA_FLASHBACK_ARCHIVE_TS describes all tablespaces in the flashback archives available in the database.

Column	Datatype	NULL	Description
FLASHBACK_ARCHIVE_NAME	VARCHAR2(255)	NOT NULL	Name of the flashback archive
FLASHBACK_ARCHIVE#	NUMBER	NOT NULL	Number of the flashback archive
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of a tablespace in the flashback archive
QUOTA_IN_MB	VARCHAR2(40)		Maximum space (in MB) that can be used for Flashback Archive from the tablespace; NULL indicates no Quota restriction

DBA_FLASHBACK_TXN_REPORT

DBA_FLASHBACK_TXN_REPORT displays information about all compensating transactions that have been committed in the database. Each row in this view is associated with one compensating transaction.

Related View

USER_FLASHBACK_TXN_REPORT displays information about the compensating transactions owned by the current user that have been committed in the database. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
COMPENSATING_XID	RAW(8)	NOT NULL	Transaction responsible for backout
COMPENSATING_TXN_NAME	VARCHAR2(256)		Name of the compensating transaction
COMMIT_TIME	DATE		Timestamp when the compensating transaction committed
XID_REPORT	CLOB		An XML report describing the details of the transactions backed out by the compensating transaction

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	User who is executing the compensating transaction

See Also: "USER_FLASHBACK_TXN_REPORT" on page 5-110

DBA_FLASHBACK_TXN_STATE

DBA_FLASHBACK_TXN_STATE displays information about the compensating status of all transactions in the database. For each compensating transaction, there could be multiple rows, where each row provides the dependency relation between the transactions that have been compensated by the compensating transaction.

Related View

USER_FLASHBACK_TXN_STATE displays information about the compensating status of the transactions owned by the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
COMPENSATING_XID	RAW(8)		Transaction ID of the compensating transaction
XID	RAW(8)		A transaction that has been compensated by the compensating transaction
DEPENDENT_XID	RAW(8)		A dependent transaction of XID **Note:** In the case of BACKOUT_MODE = CASCADE, there must be another row with XID = DEPENDENT_XID of this column.
BACKOUT_MODE	VARCHAR2(16)		Mode in which XID was backed out: ■ NOCASCADE ■ NOCASCADE_FORCE ■ NONCONFLICT_ONLY ■ CASCADE
USERNAME	VARCHAR2(30)	NOT NULL	User who is performing the compensating transaction

See Also: "USER_FLASHBACK_TXN_STATE" on page 5-110

DBA_FREE_SPACE

DBA_FREE_SPACE describes the free extents in all tablespaces in the database.

Note that if a datafile (or entire tablespace) is offline in a locally managed tablespace, you will not see any extent information. If an object has extents in an online file of the tablespace, you will see extent information about the offline datafile. However, if the object is entirely in the offline file, a query of this view will not return any records.

Related View

USER_FREE_SPACE describes the free extents in the tablespaces accessible to the current user.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the extent
FILE_ID	NUMBER		File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Starting block number of the extent

Column	Datatype	NULL	Description
BYTES	NUMBER		Size of the extent (in bytes)
BLOCKS	NUMBER		Size of the extent (in Oracle blocks)
RELATIVE_FNO	NUMBER		Relative file number of the file containing the extent

See Also: "USER_FREE_SPACE" on page 5-111

DBA_FREE_SPACE_COALESCED

DBA_FREE_SPACE_COALESCED describes statistics on coalesced space in all tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace
TOTAL_EXTENTS	NUMBER		Total number of free extents in the tablespace
EXTENTS_COALESCED	NUMBER		Total number of coalesced free extents in the tablespace
PERCENT_EXTENTS _COALESCED	NUMBER		Percentage of coalesced free extents in the tablespace
TOTAL_BYTES	NUMBER		Total number of free bytes in the tablespace
BYTES_COALESCED	NUMBER		Total number of coalesced free bytes in the tablespace
TOTAL_BLOCKS	NUMBER		Total number of free Oracle blocks in the tablespace
BLOCKS_COALESCED	NUMBER		Total number of coalesced free Oracle blocks in the tablespace
PERCENT_BLOCKS _COALESCED	NUMBER		Percentage of coalesced free Oracle blocks in the tablespace

DBA_HIGH_WATER_MARK_STATISTICS

DBA_HIGH_WATER_MARK_STATISTICS displays information about database high-watermark statistics.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Identifier of the database for which the high-watermark statistics are tracked
NAME	VARCHAR2(64)	NOT NULL	Name of the high-watermark statistic (see Table 4–2)
VERSION	VARCHAR2(17)	NOT NULL	Database version in which the high-watermarks were tracked
HIGHWATER	NUMBER		Highest value of the statistic seen at sampling time
LAST_VALUE	NUMBER		Value of the statistic at the last sample time
DESCRIPTION	VARCHAR2(128)		Description of the high-watermark statistics (see Table 4–2)

Table 4–2 DBA_HIGH_WATER_MARK_STATISTICS Statistics

Name	Description
CPU_COUNT	Maximum Number of CPUs
DATAFILES	Maximum Number of Datafiles
DB_SIZE	Maximum Size of the Database (Bytes)
INSTANCES	Oracle Database instances

Table 4–2 (Cont.) DBA_HIGH_WATER_MARK_STATISTICS Statistics

Name	Description
PART_INDEXES	Maximum Number of Partitions belonging to an User Index
PART_TABLES	Maximum Number of Partitions belonging to an User Table
QUERY_LENGTH	Maximum Query Length
SEGMENT_SIZE	Size of Largest Segment (Bytes)
SESSIONS	Maximum Number of Concurrent Sessions seen in the database
SQL_NCHAR_COLUMNS	Maximum Number of SQL NCHAR Columns
TABLESPACES	Maximum Number of Tablespaces
USER_INDEXES	Number of User Indexes
USER_MV	Maximum Number of Materialized Views (User)
USER_TABLES	Number of User Tables

DBA_HIST_ACTIVE_SESS_HISTORY

DBA_HIST_ACTIVE_SESS_HISTORY displays the history of the contents of the in-memory active session history of recent system activity. This view contains snapshots of V$ACTIVE_SESSION_HISTORY. See "V$ACTIVE_SESSION_HISTORY" on page 6-3 for further interpretation details for many of these columns (except SNAP_ID, DBID, and INSTANCE_NUMBER).

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SAMPLE_ID	NUMBER		ID of the sample
SAMPLE_TIME	TIMESTAMP(3)		Time of the sample
SESSION_ID	NUMBER		Session identifier
SESSION_SERIAL#	NUMBER		Session serial number (used to uniquely identify a session's objects)
SESSION_TYPE	VARCHAR2(10)		Session type, either FOREGROUND or BACKGROUND
USER_ID	NUMBER		Oracle user identifier
SQL_ID	VARCHAR2(13)		SQL identifier of the SQL statement that is currently being executed
SQL_CHILD_NUMBER	NUMBER		Child number of the SQL statement that is currently being executed
SQL_OPCODE	NUMBER		Indicates what phase of operation the SQL statement is in
FORCE_MATCHING_SIGNATURE	NUMBER		The signature used when the CURSOR_SHARING parameter is set to FORCE
TOP_SQL_ID	VARCHAR2(13)		???
TOP_SQL_OPCODE	NUMBER		???
SQL_PLAN_HASH_VALUE	NUMBER		Numerical representation of the SQL plan for the cursor
SQL_PLAN_LINE_ID	NUMBER		???
SQL_PLAN_OPERATION	VARCHAR2(64)		???
SQL_PLAN_OPTIONS	VARCHAR2(64)		???
SQL_EXEC_ID	NUMBER		???
SQL_EXEC_START	DATE		???

Column	Datatype	NULL	Description
PLSQL_ENTRY_OBJECT_ID	NUMBER		Object ID of top-most PL/SQL subprogram on the stack (or NULL if there is no PL/SQL subprogram on the stack).
PLSQL_ENTRY_SUBPROGRAM_ID	NUMBER		Subprogram ID of the top-most PL/SQL subprogram on the stack (or NULL if there is no PL/SQL subprogram on the stack).
PLSQL_OBJECT_ID	NUMBER		Object ID of the currently executing PL/SQL subprogram (or NULL if executing SQL)
PLSQL_SUBPROGRAM_ID	NUMBER		Subprogram ID of the currently executing PL/SQL object (or NULL if executing SQL)
QC_INSTANCE_ID	NUMBER		Query coordinator instance ID
QC_SESSION_ID	NUMBER		Query coordinator session ID
QC_SESSION_SERIAL#	NUMBER		???
EVENT	VARCHAR2(64)		If SESSION_STATE = WAITING, then the event for which the session was waiting at the time of sampling. If SESSION_STATE = ON CPU, then this column will be NULL.
EVENT_ID	NUMBER		Identifier of the resource or event for which the session is waiting or for which the session last waited
SEQ#	NUMBER		Sequence number that uniquely identifies the wait (incremented for each wait)
P1TEXT	VARCHAR2(64)		Text of first additional parameter
P1	NUMBER		First additional parameter
P2TEXT	VARCHAR2(64)		Text of second additional parameter
P2	NUMBER		Second additional parameter
P3TEXT	VARCHAR2(64)		Text of third additional parameter
P3	NUMBER		Third additional parameter
WAIT_CLASS	VARCHAR2(64)		Wait class name of the event for which the session was waiting at the time of sampling. Interpretation is similar to that of the EVENT column. Maps to V$SESSION.WAIT_CLASS.
WAIT_CLASS_ID	NUMBER		Wait class identifier of the event for which the session was waiting at the time of sampling. Interpretation is similar to that of the EVENT column. Maps to V$SESSION.WAIT_CLASS_ID.
WAIT_TIME	NUMBER		Total wait time for the event for which the session last waited (0 if currently waiting)
SESSION_STATE	VARCHAR2(7)		Session state: ■ WAITING ■ ON CPU
TIME_WAITED	NUMBER		Time that the current session actually spent waiting for the event. This column is set for waits that were in progress at the time the sample was taken.
BLOCKING_SESSION_STATUS	VARCHAR2(11)		Status of the blocking session: ■ VALID ■ NO HOLDER ■ GLOBAL ■ NOT IN WAIT ■ UNKNOWN
BLOCKING_SESSION	NUMBER		Session identifier of the blocking session. Populated only when the session was waiting for enqueues or a "buffer busy" wait. Maps to V$SESSION.BLOCKING_SESSION.

Column	Datatype	NULL	Description
BLOCKING_SESSION_SERIAL#	NUMBER		Serial number of the blocking session
CURRENT_OBJ#	NUMBER		Object ID of the object that the session is currently referencing. This information is only available if the session was waiting for Application, Cluster, Concurrency, and User I/O wait events. Maps to V$SESSION.ROW_WAIT_OBJ#.
CURRENT_FILE#	NUMBER		File number of the file containing the block that the session is currently referencing. This information is only available if the session was waiting for Cluster, Concurrency, and User I/O wait events. Maps to V$SESSION.ROW_WAIT_FILE#.
CURRENT_BLOCK#	NUMBER		ID of the block that the session is currently referencing
CURRENT_ROW#	NUMBER		???
CONSUMER_GROUP_ID	NUMBER		???
XID	RAW(8)		Transaction ID that the session was working on at the time of sampling. V$SESSION does not contain this information.
REMOTE_INSTANCE#	NUMBER		Remote instance identifier that will serve the block that this session is waiting for. This information is only available if the session was waiting for cluster events.
IN_CONNECTION_MGMT	VARCHAR2(1)		Indicates whether the session was doing connection management at the time of sampling (Y) or not (N)
IN_PARSE	VARCHAR2(1)		Indicates whether the session was parsing at the time of sampling (Y) or not (N)
IN_HARD_PARSE	VARCHAR2(1)		Indicates whether the session was hard parsing at the time of sampling (Y) or not (N)
IN_SQL_EXECUTION	VARCHAR2(1)		Indicates whether the session was executing SQL statements at the time of sampling (Y) or not (N)
IN_PLSQL_EXECUTION	VARCHAR2(1)		Indicates whether the session was executing PL/SQL at the time of sampling (Y) or not (N)
IN_PLSQL_RPC	VARCHAR2(1)		Indicates whether the session was executing inbound PL/SQL RPC calls at the time of sampling (Y) or not (N)
IN_PLSQL_COMPILATION	VARCHAR2(1)		Indicates whether the session was compiling PL/SQL at the time of sampling (Y) or not (N)
IN_JAVA_EXECUTION	VARCHAR2(1)		Indicates whether the session was executing Java at the time of sampling (Y) or not (N)
IN_BIND	VARCHAR2(1)		Indicates whether the session was doing bind operations at the time of sampling (Y) or not (N)
IN_CURSOR_CLOSE	VARCHAR2(1)		Indicates whether the session was closing a cursor at the time of sampling (Y) or not (N)
SERVICE_HASH	NUMBER		Hash that identifies the Service
PROGRAM	VARCHAR2(48)		Name of the operating system program
MODULE	VARCHAR2(48)		Name of the currently executing module as set by the DBMS_APPLICATION_INFO.SET_MODULE procedure
ACTION	VARCHAR2(32)		Name of the currently executing action as set by the DBMS_APPLICATION_INFO.SET_ACTION procedure
CLIENT_ID	VARCHAR2(64)		Client identifier of the session

DBA_HIST_BASELINE

DBA_HIST_BASELINE displays information on baselines taken in the system. For each baseline, this view displays the complete time range and whether the baseline is the default baseline.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
BASELINE_ID	NUMBER	NOT NULL	Internal ID for the baseline
BASELINE_NAME	VARCHAR2(64)		User-specified name for the baseline
BASELINE_TYPE	VARCHAR2(13)		The baseline type, as follows: STATIC - baselines that are created manually by the user MOVING WINDOW - baselines that have dynamic start and end snapshot IDs GENERATED - baselines that are automatically generated by the system, using a template
START_SNAP_ID	NUMBER		Start snapshot ID for the baseline
START_SNAP_TIME	TIMESTAMP(3)		Time associated with the start snapshot ID
END_SNAP_ID	NUMBER		End snapshot ID for the baseline
END_SNAP_TIME	TIMESTAMP(3)		Time associated with the end snapshot ID
MOVING_WINDOW_SIZE	NUMBER		If BASELINE_TYPE is MOVING WINDOW, this field is the size of the moving window in number of days. If NULL, then the window size is the value of the AWR retention setting.
CREATION_TIME	DATE		Time the baseline was created
EXPIRATION	NUMBER		How long to keep the baseline, in number of days. A NULL value means that the baseline will be kept forever.
TEMPLATE_NAME	VARCHAR2(64)		Name of the template that created this baseline, if any.
LAST_TIME_COMPUTED	DATE		Last time that statistics were computed on the baseline.

DBA_HIST_BASELINE_DETAILS

DBA_HIST_BASELINE_DETAILS displays details about the baseline.

Column	Datatype	NULL	Description
DBID	NUMBER		Database ID
INSTANCE_NUMBER	NUMBER		Instance ID for the baseline data
BASELINE_ID	NUMBER		Internal ID for the baseline
BASELINE_NAME	VARCHAR2(64)		User-specified name for the baseline
BASELINE_TYPE	VARCHAR2(13)		The baseline type, as follows: STATIC - baselines that are created manually by the user MOVING WINDOW - baselines that have dynamic start and end snapshot IDs GENERATED - baselines that are automatically generated by the system, using a template
START_SNAP_ID	NUMBER		Start snapshot ID for the baseline
START_SNAP_TIME	TIMESTAMP(3)		Start snapshot time for the baseline
END_SNAP_ID	NUMBER		End snapshot ID for the baseline
END_SNAP_TIME	TIMESTAMP(3)		End snapshot time for the baseline
SHUTDOWN	VARCHAR2(3)		Whether or not there is a database startup or shutdown in this interval (YES, NO, or NULL).
ERROR_COUNT	NUMBER		Number of errors in the snapshots in the baseline snapshot range
PCT_TOTAL_TIME	NUMBER		Amount of time captured in snapshots, divided by the total possible time for this baseline

Column	Datatype	NULL	Description
LAST_TIME_COMPUTED	DATE		Last time that statistics were computed on the baseline
MOVING_WINDOW_SIZE	NUMBER		If BASELINE_TYPE is MOVING WINDOW, this field is the size of the moving window in number of days. If NULL, then the window size is the value of the AWR retention setting.
CREATION_TIME	DATE		Time the baseline was created
EXPIRATION	NUMBER		How long to keep the baseline, in number of days. A value of NULL indicates that the baseline will be kept forever.
TEMPLATE_NAME	VARCHAR2(64)		Name of the template that created this baseline, if any.

DBA_HIST_BASELINE_METADATA

DBA_HIST_BASELINE_METADATA displays metadata information for the baseline.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
BASELINE_ID	NUMBER	NOT NULL	Internal ID for the baseline
BASELINE_NAME	VARCHAR2(64)		User-specified name for the baseline
BASELINE_TYPE	VARCHAR2(13)		The baseline type, as follows: STATIC - baselines that are created manually by the user MOVING WINDOW - baselines that have dynamic start and end snapshot IDs GENERATED - baselines that are automatically generated by the system, using a template
START_SNAP_ID	NUMBER		Start snapshot ID for the baseline
END_SNAP_ID	NUMBER		End snapshot ID for the baseline
MOVING_WINDOW_SIZE	NUMBER		If BASELINE_TYPE is MOVING WINDOW, this field is the size of the moving window in number of days. If NULL, then the window size is the value of the AWR retention setting.
CREATION_TIME	DATE		Time the baseline was created
EXPIRATION	NUMBER		How long to keep the baseline, in number of days. If the value is NULL, the baseline will be kept forever.
TEMPLATE_NAME	VARCHAR2(64)		Name of the template that created this baseline, if any
LAST_TIME_COMPUTED	DATE		Last time that statistics were computed on the baseline

DBA_HIST_BASELINE_TEMPLATE

DBA_HIST_BASELINE_TEMPLATE displays the templates used by the system for baseline generation. The system uses this information to determine which baselines should be automatically created or removed.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
TEMPLATE_ID	NUMBER	NOT NULL	Internal ID for the template
TEMPLATE_NAME	VARCHAR2(30)	NOT NULL	Name of the template

Column	Datatype	NULL	Description
TEMPLATE_TYPE	VARCHAR2(9)	NOT NULL	Type of the template, as follows: SINGLE - one time period REPEATING - maintain a time period
BASELINE_NAME_PREFIX	VARCHAR2(30)	NOT NULL	Name to use for the baselines that are created: For a template type of SINGLE, the BASELINE_NAME_PREFIX is the name that will be used. For a template type of REPEATING, the BASELINE_NAME will be the prefix to the name.
START_TIME	DATE	NOT NULL	For a template type of SINGLE, this is the start time for future baselines For a template type of REPEATING, this is the effective start time that baselines should start being generated.
END_TIME	DATE	NOT NULL	For a template type of SINGLE, this is the end time for future baselines. For a template type of REPEATING, this is the effective end time that baselines should stop being generated.
DAY_OF_WEEK	VARCHAR2(9)		For a template type of REPEATING, this indicates the day of the week to create the baseline: SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, ALL.
HOUR_IN_DAY	NUMBER		For a template type of REPEATING, a value from 0 - 23 to indicate the hour of the day to create the baseline for.
DURATION	NUMBER		For a template type of REPEATING, the length of time for the baseline to be created.
EXPIRATION	NUMBER		How long to keep the baseline, in number of days
REPEAT_INTERVAL	VARCHAR2(128)		String that represents the time repeating information in the format used by the DBMS_SCHEDULER package
LAST_GENERATED	DATE		Last time a baseline was generated for this template

DBA_HIST_BG_EVENT_SUMMARY

DBA_HIST_BG_EVENT_SUMMARY displays the historical summary background event activity. This view contains snapshots from V$SESSION_EVENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2(64)	NOT NULL	Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name of the class of the wait event
TOTAL_WAITS	NUMBER		Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER		Total number of timeouts for the event
TIME_WAITED_MICRO	NUMBER		Total amount of time waited for the event (in microseconds)

DBA_HIST_BUFFER_POOL_STAT

DBA_HIST_BUFFER_POOL_STAT displays historical statistics about all buffer pools available for the instance. This view contains snapshots of V$BUFFER_POOL_STATISTICS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
ID	NUMBER	NOT NULL	Buffer pool identifier number
NAME	VARCHAR2(20)		Name of the buffer pool
BLOCK_SIZE	NUMBER		Block Size
SET_MSIZE	NUMBER		Buffer pool maximum set size
CNUM_REPL	NUMBER		Number of buffers on the replacement list
CNUM_WRITE	NUMBER		Number of buffers on the write list
CNUM_SET	NUMBER		Number of buffers in the set
BUF_GOT	NUMBER		Number of buffers gotten by the set
SUM_WRITE	NUMBER		Number of buffers written by the set
SUM_SCAN	NUMBER		Number of buffers scanned in the set
FREE_BUFFER_WAIT	NUMBER		Free buffer wait statistic
WRITE_COMPLETE_WAIT	NUMBER		Write complete wait statistic
BUFFER_BUSY_WAIT	NUMBER		Buffer busy wait statistic
FREE_BUFFER_INSPECTED	NUMBER		Free buffer inspected statistic
DIRTY_BUFFERS_INSPECTED	NUMBER		Dirty buffers inspected statistic
DB_BLOCK_CHANGE	NUMBER		Database blocks changed statistic
DB_BLOCK_GETS	NUMBER		Database blocks gotten statistic
CONSISTENT_GETS	NUMBER		Consistent gets statistic
PHYSICAL_READS	NUMBER		Physical reads statistic
PHYSICAL_WRITES	NUMBER		Physical writes statistic

DBA_HIST_BUFFERED_QUEUES

DBA_HIST_BUFFERED_QUEUES displays historical information about all buffered queues available for the instance.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
QUEUE_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
STARTUP_TIME	DATE	NOT NULL	Startup time of the instance
QUEUE_ID	NUMBER	NOT NULL	ID of the queue
NUM_MSGS	NUMBER		Total number of outstanding messages currently enqueued in the buffered queue for the subscriber (includes the count of the messages overflowed to disk)

Column	Datatype	NULL	Description
SPILL_MSGS	NUMBER		Current number of overflow messages spilled to disk from the buffered queue
CNUM_MSGS	NUMBER		Cumulative total number of messages enqueued into the buffered queue since the buffered queue was created.
CSPILL_MSGS	NUMBER		Cumulative total number of overflow messages spilled to disk from the buffered queue since the buffered queue was created

DBA_HIST_BUFFERED_SUBSCRIBERS

DBA_HIST_BUFFERED_SUBSCRIBERS displays historical information about the subscribers for all buffered queues in the instance.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
QUEUE_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
SUBSCRIBER_ID	NUMBER	NOT NULL	Internal subscriber number (for identification)
SUBSCRIBER_NAME	VARCHAR2(30)		Name of the subscriber
SUBSCRIBER_ADDRESS	VARCHAR2(1024)		Address of the subscribing agent
SUBSCRIBER_TYPE	VARCHAR2(30)		Type of subscriber: ■ PROXY - Proxy subscriber ■ SUBSCRIBOR
STARTUP_TIME	DATE	NOT NULL	Startup time of the instance
NUM_MSGS	NUMBER		Total number of outstanding messages currently enqueued in the buffered queue for the subscriber (includes the count of the messages overflowed to disk)
CNUM_MSGS	NUMBER		Cumulative total number of messages enqueued for the subscriber since the creation of the buffered queue
TOTAL_SPILLED_MSG	NUMBER		Total number of spilled messages for the subscriber

DBA_HIST_CLUSTER_INTERCON

DBA_HIST_CLUSTER_INTERCON displays information about the devices used by the instance to access the interconnect (that is, communicate with other instances).

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAME	VARCHAR2(15)	NOT NULL	Operating system name of the device
IP_ADDRESS	VARCHAR2(16)		IP address of the device
IS_PUBLIC	VARCHAR2(3)		Indicates whether the device is a public interface (YES) or a private interface (NO) Public interfaces can be listened to by outside applications, which may be a security problem. Oracle recommends using private interfaces for interconnect.

Column	Datatype	NULL	Description
SOURCE	VARCHAR2(31)		Describes the type of device

DBA_HIST_COLORED_SQL

DBA_HIST_COLORED_SQL displays the SQL IDs that have been marked for AWR SQL capture. If a SQL statement is colored using the DBMS_WORKLOAD_REPOSITORY.ADD_COLORED_SQL procedure, then AWR will always capture the SQL statistics for the colored SQL ID. A SQL statement can be removed from coloring using the DBMS_WORKLOAD_REPOSITORY.REMOVE_COLORED_SQL procedure.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SQL_ID	VARCHAR2(13)	NOT NULL	SQL ID of colored SQL statement
CREATE_TIME	DATE	NOT NULL	Time the SQL ID was colored

DBA_HIST_COMP_IOSTAT

DBA_HIST_COMP_IOSTAT displays information about I/O statistics aggregated on the component level.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
COMPONENT	VARCHAR2(64)	NOT NULL	Component name
FILE_TYPE	VARCHAR2(64)	NOT NULL	File type
IO_TYPE	CHAR(5)	NOT NULL	The type of I/O performed
OPERATION	CHAR(5)	NOT NULL	Operation name
BYTES	NUMBER	NOT NULL	Number of bytes
IO_COUNT	NUMBER	NOT NULL	Number of I/Os that were performed

DBA_HIST_CR_BLOCK_SERVER

DBA_HIST_CR_BLOCK_SERVER displays historical statistics on the Global Cache Service processes (lMS) used in cache fusion. This view contains snapshots of V$CR_BLOCK_SERVER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
CR_REQUESTS	NUMBER		Number of CR blocks served due to remote CR block requests
CURRENT_REQUESTS	NUMBER		Number of current blocks served due to remote CR block requests CR_REQUESTS + CURRENT_REQUESTS = global cache cr clocks served (from V$SYSSTAT).
DATA_REQUESTS	NUMBER		Number of current or CR requests for data blocks

Column	Datatype	NULL	Description
UNDO_REQUESTS	NUMBER		Number of CR requests for undo blocks
TX_REQUESTS	NUMBER		Number of CR requests for undo segment header blocks DATA_REQUESTS + UNDO_REQUESTS + TX_REQUESTS = total number of requests handled by the LMS processes
CURRENT_RESULTS	NUMBER		Number of requests for which no changes were rolled out of the block returned to the requesting instance
PRIVATE_RESULTS	NUMBER		Number of requests for which changes were rolled out of the block returned to the requesting instance, and only the requesting transaction can use the resulting CR block
ZERO_RESULTS	NUMBER		Number of requests for which changes were rolled out of the block returned to the requesting instance. Only zero-XID transactions can use the block.
DISK_READ_RESULTS	NUMBER		Number of requests for which the requesting instance had to read the requested block from disk
FAIL_RESULTS	NUMBER		Number of requests that failed; the requesting transaction must reissue the request
FAIRNESS_DOWN_CONVERTS	NUMBER		Number of times an instance receiving a request has down-converted an X lock on a block because it was not modifying the block
FAIRNESS_CLEARS	NUMBER		Number of times the fairness counter was cleared. This counter tracks the number of times a block was modified after it was served.
FREE_GC_ELEMENTS	NUMBER		Number of times a request was received from another instance and the X lock had no buffers
FLUSHES	NUMBER		Number of times the log has been flushed by an LMS process
FLUSHES_QUEUED	NUMBER		Number of flushes queued by an LMS process
FLUSH_QUEUE_FULL	NUMBER		Number of times the flush queue was full
FLUSH_MAX_TIME	NUMBER		Maximum time for flush
LIGHT_WORKS	NUMBER		Number of times the light-work rule was evoked. This rule prevents the LMS processes from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the LMS process from completing its response to the CR request.
ERRORS	NUMBER		Number of times an error was signalled by an LMS process

DBA_HIST_CURRENT_BLOCK_SERVER

DBA_HIST_CURRENT_BLOCK_SERVER displays historical statistics on the Global Cache Service processes (lMS) used in cache fusion. This view contains snapshots of V$CURRENT_BLOCK_SERVER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
PIN1	NUMBER		Pins taking less than 1 millisecond
PIN10	NUMBER		Pins taking 1 to 10 milliseconds
PIN100	NUMBER		Pins taking 10 to 100 milliseconds
PIN1000	NUMBER		Pins taking 100 to 1000 milliseconds

Column	Datatype	NULL	Description
PIN10000	NUMBER		Pins taking 1000 to 10000 milliseconds
FLUSH1	NUMBER		Flushes taking less than 1 millisecond
FLUSH10	NUMBER		Flushes taking 1 to 10 milliseconds
FLUSH100	NUMBER		Flushes taking 10 to 100 milliseconds
FLUSH1000	NUMBER		Flushes taking 100 to 1000 milliseconds
FLUSH10000	NUMBER		Flushes taking 1000 to 10000 milliseconds
WRITE1	NUMBER		Writes taking less than 1 millisecond
WRITE10	NUMBER		Writes taking 1 to 10 milliseconds
WRITE100	NUMBER		Writes taking 10 to 100 milliseconds
WRITE1000	NUMBER		Writes taking 100 to 1000 milliseconds
WRITE10000	NUMBER		Writes taking 1000 to 10000 milliseconds

DBA_HIST_DATABASE_INSTANCE

DBA_HIST_DATABASE_INSTANCE displays the databases and instances in the Workload Repository.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number
STARTUP_TIME	TIMESTAMP(3)	NOT NULL	Startup time of the instance
PARALLEL	VARCHAR2(3)	NOT NULL	Indicates whether the instance is running in a Real Application Clusters (RAC) environment (YES) or not (NO)
VERSION	VARCHAR2(17)	NOT NULL	Database version
DB_NAME	VARCHAR2(9)		Name of the database
INSTANCE_NAME	VARCHAR2(16)		Name of the instance
HOST_NAME	VARCHAR2(64)		Name of the host
LAST_ASH_SAMPLE_ID	NUMBER	NOT NULL	Last sample ID for the active session history

DBA_HIST_DATAFILE

DBA_HIST_DATAFILE displays a history of the datafile information from the control file. This view contains snapshots of V$DATAFILE.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the datafile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the datafile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the datafile

DBA_HIST_DB_CACHE_ADVICE

DBA_HIST_DB_CACHE_ADVICE displays historical predictions of the number of physical reads for the cache size corresponding to each row. This view contains snapshots of V$DB_CACHE_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
BPID	NUMBER		Buffer Pool identifier (ranges from 1 to 8)
BUFFERS_FOR_ESTIMATE	NUMBER		Cache size for prediction (in terms of buffers)
NAME	VARCHAR2(20)		Buffer pool name
BLOCK_SIZE	NUMBER		Block size in bytes for buffers in the pool (the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, or 32768)
ADVICE_STATUS	VARCHAR2(3)		Status of the advisory: ■ ON - Currently running ■ OFF - Disabled (the estimates are historical and calculated when last enabled)
SIZE_FOR_ESTIMATE	NUMBER		Cache size for prediction (in megabytes)
SIZE_FACTOR	NUMBER		Size factor with respect to the current cache size
PHYSICAL_READS	NUMBER		Physical reads for the cache size
BASE_PHYSICAL_READS	NUMBER		Base physical reads for the cache size
ACTUAL_PHYSICAL_READS	NUMBER		Actual physical reads for the cache size

DBA_HIST_DLM_MISC

DBA_HIST_DLM_MISC displays miscellaneous Real Application Clusters (RAC) statistics. This view contains snapshots of V$DLM_MISC.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STATISTIC#	NUMBER		Statistic number
NAME	VARCHAR2(38)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_ENQUEUE_STAT

DBA_HIST_ENQUEUE_STAT displays historical statistics on the number of enqueue (lock) requests for each type of lock. This view contains snapshots of V$ENQUEUE_STATISTICS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot

Column	Datatype	NULL	Description
EQ_TYPE	VARCHAR2(2)	NOT NULL	Type of enqueue requested
REQ_REASON	VARCHAR2(64)	NOT NULL	Reason for the enqueue request
TOTAL_REQ#	NUMBER		Total number of enqueue requests or enqueue conversions for this type of enqueue
TOTAL_WAIT#	NUMBER		Total number of times an enqueue request or conversion resulted in a wait
SUCC_REQ#	NUMBER		Number of times an enqueue request or conversion was granted
FAILED_REQ#	NUMBER		Number of times an enqueue request or conversion failed
CUM_WAIT_TIME	NUMBER		Total amount of time (in milliseconds) spent waiting for the enqueue or enqueue conversion
EVENT#	NUMBER		Event number

DBA_HIST_EVENT_HISTOGRAM

DBA_HIST_EVENT_HISTOGRAM displays event histogram historical statistics information. This view contains snapshots of V$EVENT_HISTOGRAM.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2(64)	NOT NULL	Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name of the class of the wait event
WAIT_TIME_MILLI	NUMBER	NOT NULL	Wait time (in milliseconds)
WAIT_COUNT	NUMBER		Wait count

DBA_HIST_EVENT_NAME

DBA_HIST_EVENT_NAME displays information about wait events. This view contains a snapshot of V$EVENT_NAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2(64)	NOT NULL	Name of the wait event
PARAMETER1	VARCHAR2(64)		Description of the first parameter for the wait event
PARAMETER2	VARCHAR2(64)		Description of the second parameter for the wait event
PARAMETER3	VARCHAR2(64)		Description of the third parameter for the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name of the class of the wait event

DBA_HIST_FILEMETRIC_HISTORY

DBA_HIST_FILEMETRIC_HISTORY displays the history of file metrics collected in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FILEID	NUMBER	NOT NULL	File number
CREATIONTIME	NUMBER	NOT NULL	File creation time
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	ID of the group to which the file belongs
AVGREADTIME	NUMBER	NOT NULL	Average file read time
AVGWRITETIME	NUMBER	NOT NULL	Average file write time
PHYSICALREAD	NUMBER	NOT NULL	Number of physical reads
PHYSICALWRITE	NUMBER	NOT NULL	Number of physical writes
PHYBLKREAD	NUMBER	NOT NULL	Number of physical block reads
PHYBLKWRITE	NUMBER	NOT NULL	Number of physical block writes

DBA_HIST_FILESTATXS

DBA_HIST_FILESTATXS displays information about file read/write statistics. This view contains snapshots of V$FILESTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
FILE#	NUMBER		File identification number
CREATION_CHANGE#	NUMBER		Change number at which the datafile was created
FILENAME	VARCHAR2(513)		Name of the datafile
TS#	NUMBER		Tablespace number
TSNAME	VARCHAR2(30)		Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the datafile
PHYRDS	NUMBER		Number of physical reads done
PHYWRTS	NUMBER		Number of times DBWR is required to write
SINGLEBLKRDS	NUMBER		Number of single block reads
READTIM	NUMBER		Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if TIMED_STATISTICS is false
WRITETIM	NUMBER		Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if TIMED_STATISTICS is false
SINGLEBLKRDTIM	NUMBER		Cumulative single block read time (in hundredths of a second)

Column	Datatype	NULL	Description
PHYBLKRD	NUMBER		Number of physical blocks read
PHYBLKWRT	NUMBER		Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
WAIT_COUNT	NUMBER		Wait Count
TIME	NUMBER		Wait Time

DBA_HIST_IC_CLIENT_STATS

DBA_HIST_IC_CLIENT_STATS displays information about the usage of an interconnect device by the instance. The information is divided into several areas of the Oracle Database, each identified by the NAME value.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAME	VARCHAR2(9)	NOT NULL	Identifies the area of the Oracle Database: ■ ipq - Parallel query communications ■ dlm - Database lock management ■ cache - Global cache communications All other values are internal to Oracle and are not expected to have high usage.
BYTES_SENT	NUMBER		Number of bytes sent by the instance since instance startup for the software area identified by NAME. This information is aggregated across all devices used by the instance.
BYTES_RECEIVED	NUMBER		Number of bytes received by the instance since instance startup for the software area identified by NAME. This information is aggregated across all devices used by the instance.

DBA_HIST_IC_DEVICE_STATS

DBA_HIST_IC_DEVICE_STATS displays operating system information about the usage of interconnect devices by the machine. This usage contains Oracle usage but is not limited to it. The quality of the information depends on the operating system.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
IF_NAME	VARCHAR2(15)	NOT NULL	Name of the device (same as NAME in DBA_HIST_CLUSTER_INTERCON)
IP_ADDR	VARCHAR2(16)		IP address of the device (same as IP_ADDRESS in DBA_HIST_CLUSTER_INTERCON)
NET_MASK	VARCHAR2(16)		Network mask
FLAGS	VARCHAR2(32)		Flags
MTU	NUMBER		Maximum transmission unit
BYTES_RECEIVED	NUMBER		Number of bytes received since OS start time
PACKETS_RECEIVED	NUMBER		Number of packets received since OS start time

Column	Datatype	NULL	Description
RECEIVE_ERRORS	NUMBER		Number of receive errors since OS start time
RECEIVE_DROPPED	NUMBER		Number of receive messages that were dropped
RECEIVE_BUF_OR	NUMBER		Number of receive buffer overruns experienced
RECEIVE_FRAME_ERR	NUMBER		Number of receive errors due to frame error
BYTES_SENT	NUMBER		Number of bytes sent since OS start time
PACKETS_SENT	NUMBER		Number of packets sent since OS start time
SEND_ERRORS	NUMBER		Number of send errors since OS start time
SENDS_DROPPED	NUMBER		Number of send messages that were dropped
SEND_BUF_OR	NUMBER		Number of send buffer overruns experienced
SEND_CARRIER_LOST	NUMBER		Number of send errors due to carrier lost

DBA_HIST_INST_CACHE_TRANSFER

DBA_HIST_INST_CACHE_TRANSFER displays the historical statistics on the cache blocks transferred among instances. This view contains snapshots of V$INSTANCE_CACHE_TRANSFER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
INSTANCE	NUMBER		Instance from which the blocks are transferred
CLASS	VARCHAR2(18)		Class of the cache block
CR_BLOCK	NUMBER		CR block transfers not affected by remote processing delays
CR_BUSY	NUMBER		Current block transfers affected by remote contention
CR_CONGESTED	NUMBER		CR block transfers affected by remote system load
CURRENT_BLOCK	NUMBER		Current block transfers not affected by remote processing delays
CURRENT_BUSY	NUMBER		Current block transfers affected by remote contention
CURRENT_CONGESTED	NUMBER		Current block transfers affected by remote system load
LOST	NUMBER		???
CR_2HOP	NUMBER		???
CR_3HOP	NUMBER		???
CURRENT_2HOP	NUMBER		???
CURRENT_3HOP	NUMBER		???
CR_BLOCK_TIME	NUMBER		???
CR_BUSY_TIME	NUMBER		???
CR_CONGESTED_TIME	NUMBER		???
CURRENT_BLOCK_TIME	NUMBER		???
CURRENT_BUSY_TIME	NUMBER		???
CURRENT_CONGESTED_TIME	NUMBER		???
LOST_TIME	NUMBER		???
CR_2HOP_TIME	NUMBER		???

Column	Datatype	NULL	Description
CR_3HOP_TIME	NUMBER		???
CURRENT_2HOP_TIME	NUMBER		???
CURRENT_SHOP_TIME	NUMBER		???

DBA_HIST_INSTANCE_RECOVERY

DBA_HIST_INSTANCE_RECOVERY displays the historical monitoring of the mechanisms available to the user to limit recovery I/O. This view contains snapshots of V$INSTANCE_RECOVERY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
RECOVERY_ESTIMATED_IOS	NUMBER		Number of dirty buffers in the buffer cache. In the Standard Edition, this column is always null.
ACTUAL_REDO_BLKS	NUMBER		Current actual number of redo blocks required for recovery
TARGET_REDO_BLKS	NUMBER		Current target number of redo blocks that must be processed for recovery. This value is the minimum value of the following 3 columns, and identifies which of the 3 user-defined limits determines checkpointing.
LOG_FILE_SIZE_REDO_BLKS	NUMBER		Maximum number of redo blocks required to guarantee that a log switch does not occur before the checkpoint completes
LOG_CHKPT_TIMEOUT_REDO_BLKS	NUMBER		Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_TIMEOUT parameter. The value displayed is not meaningful unless LOG_CHECKPOINT_TIMEOUT has been set.
LOG_CHKPT_INTERVAL_REDO_BLKS	NUMBER		Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_INTERVAL parameter. The value displayed is not meaningful unless LOG_CHECKPOINT_INTERVAL has been set.
FAST_START_IO_TARGET_REDO_BLKS	NUMBER		This column is obsolete and maintained for backward compatibility. The value of this column is always null.
TARGET_MTTR	NUMBER		Effective MTTR (mean time to recover) target value in seconds. The TARGET_MTTR value is calculated based on the value of the FAST_START_MTTR_TARGET parameter (the TARGET_MTTR value is used internally), and is usually an approximation of the parameter's value. However, if the FAST_START_MTTR_TARGET parameter value is very small (for example, one second), or very large (for example, 3600 seconds), then the calculation will produce a target value dictated by system limitations. In such cases, the TARGET_MTTR value will be the shortest calculated time, or the longest calculated time that recovery is expected to take. If FAST_START_MTTR_TARGET is not specified, then the value of this field is the current estimated MTTR.
ESTIMATED_MTTR	NUMBER		Current estimated mean time to recover (MTTR) based on the number of dirty buffers and log blocks (0 if FAST_START_MTTR_TARGET is not specified). This value tells you how long you can expect recovery to take based on the work the system is doing right now.
CKPT_BLOCK_WRITES	NUMBER		Number of blocks written by checkpoint writes

Column	Datatype	NULL	Description
OPTIMAL_LOGFILE_SIZE	NUMBER		Redo log file size (in megabytes) that is considered optimal based on the current setting of FAST_START_MTTR_TARGET. It is recommended that all online redo logs be configured to be at least this value.
ESTD_CLUSTER_AVAILABLE_TIME	NUMBER		Estimated time (in seconds) that the cluster would become partially available should the instance fail. This column is only meaningful in a Real Application Clusters (RAC) environment. In a non-RAC environment, the value of this column is null.
WRITES_MTTR	NUMBER		Number of writes driven by the FAST_START_MTTR_TARGET parameter
WRITES_LOGFILE_SIZE	NUMBER		Number of writes driven by the smallest redo log file size
WRITES_LOG_CHECKPOINT_SETTINGS	NUMBER		Number of writes driven by the LOG_CHECKPOINT_INTERVAL parameter or the LOG_CHECKPOINT_TIMEOUT parameter
WRITES_OTHER_SETTINGS	NUMBER		Number of writes driven by other reasons (such as the deprecated FAST_START_IO_TARGET parameter)
WRITES_AUTOTUNE	NUMBER		Number of writes due to auto-tune checkpointing
WRITES_FULL_THREAD_CKPT	NUMBER		Number of writes due to full thread checkpoints

DBA_HIST_INTERCONNECT_PINGS

DBA_HIST_INTERCONNECT_PINGS displays information about measured latency of interconnect messages (round-trip) from instance to instance. In Oracle Database 11*g*, the PING process assesses the latencies associated with communications for each pair of instances.

Every few seconds, the process in one instance (INSTANCE_NUMBER value) sends two messages to each instance (TARGET_INSTANCE value). One message has a size of 500 bytes and the other has a size of 8 KB. The message is received by the PING process on the target instance and is immediately acknowledged. The time for the round-trip is measured and collected.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
TARGET_INSTANCE	NUMBER	NOT NULL	Target instance number
CNT_500B	NUMBER		Number of pings of size 500 bytes from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER)
WAIT_500B	NUMBER		Sum of round-trip times for messages of size 500 bytes from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER). Dividing by CNT_500B gives the average latency.
WAITSQ_500B	NUMBER		Sum of squares (divided by 1000) of round-trip times for messages of size 500 bytes from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER). When used with CNT_500B and WAIT_500B, the standard deviation of the latency can be calculated.
CNT_8K	NUMBER		Number of pings of size 8 KB from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER)

Column	Datatype	NULL	Description
WAIT_8K	NUMBER		Sum of round-trip times for messages of size 8 KB from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER). Dividing by CNT_8K gives the average latency.
WAITSQ_8K	NUMBER		Sum of squares (divided by 1000) of round-trip times for messages of size 8 KB from INSTANCE_NUMBER to TARGET_INSTANCE since the startup of the source instance (INSTANCE_NUMBER). When used with CNT_8K and WAIT_8K, the standard deviation of the latency can be calculated.

DBA_HIST_IOSTAT_FILETYPE

DBA_HIST_IOSTAT_FILETYPE displays historical I/O statistics by file type. This view contains snapshots of V$IOSTAT_FILE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FILETYPE_ID	NUMBER	NOT NULL	Type of file (for example, log file, data file, and so on)
FILETYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the file, in the case of a data file or temp file. For all other files, a corresponding string to be displayed (for example, ARCHIVELOG).
SMALL_READ_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes written
SMALL_READ_REQS	NUMBER	NOT NULL	Number of single block read requests
SMALL_WRITE_REQS	NUMBER	NOT NULL	Number of single block write requests
SMALL_SYNC_READ_REQS	NUMBER	NOT NULL	Number of synchronous single block read requests
LARGE_READ_REQS	NUMBER	NOT NULL	Number of multiblock read requests
LARGE_WRITE_REQS	NUMBER	NOT NULL	Number of multiblock write requests
SMALL_READ_SERVICETIME	NUMBER	NOT NULL	Total service time (in milliseconds) for single block read requests
SMALL_WRITE_SERVICETIME	NUMBER	NOT NULL	Total service time (in milliseconds) for single block write requests
SMALL_SYNC_READ_LATENCY	NUMBER	NOT NULL	Latency for single block synchronous reads (in milliseconds)
LARGE_READ_SERVICETIME	NUMBER	NOT NULL	Total service time (in milliseconds) for multiblock read requests
LARGE_WRITE_SERVICETIME	NUMBER	NOT NULL	Total service time (in milliseconds) for multiblock write requests
RETRIES_ON_ERROR	NUMBER	NOT NULL	Number of read retries on error

DBA_HIST_IOSTAT_FILETYPE_NAME

DBA_HIST_IOSTAT_FILETYPE_NAME displays historical I/O statistics for file type names. This view contains snapshots of V$IOSTAT_FILE.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
FILETYPE_ID	NUMBER	NOT NULL	Type of file (for example, log file, data file, and so on)
FILETYPE_NAME	VARCHAR2(30)	NOT NULL	Name of the file, in the case of a data file or temp file. For all other files, a corresponding string to be displayed (for example, ARCHIVELOG).

DBA_HIST_IOSTAT_FUNCTION

DBA_HIST_IOSTAT_FUNCTION displays historical I/O statistics by function. This view contains snapshots of V$IOSTAT_FUNCTION.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FUNCTION_ID	NUMBER	NOT NULL	Function ID
FUNCTION_NAME	VARCHAR2(18)	NOT NULL	Function name
SMALL_READ_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes written
SMALL_READ_REQS	NUMBER	NOT NULL	Number of single block read requests
SMALL_WRITE_REQS	NUMBER	NOT NULL	Number of single block write requests
LARGE_READ_REQS	NUMBER	NOT NULL	Number of multiblock read requests
LARGE_WRITE_REQS	NUMBER	NOT NULL	Number of multiblock write requests
NUMBER_OF_WAITS	NUMBER	NOT NULL	Number of I/O waits by functionality
WAIT_TIME	NUMBER	NOT NULL	Total wait time (in milliseconds)

DBA_HIST_IOSTAT_FUNCTION_NAME

DBA_HIST_IOSTAT_FUNCTION_NAME displays historical I/O statistics by function names. This view contains snapshots of V$IOSTAT_FUNCTION.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
FUNCTION_ID	NUMBER	NOT NULL	Function ID
FUNCTION_NAME	VARCHAR2(18)	NOT NULL	Function name

DBA_HIST_JAVA_POOL_ADVICE

DBA_HIST_JAVA_POOL_ADVICE displays historical information about estimated parse time in the Java pool for different pool sizes. This view contains snapshots of V$JAVA_POOL_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
JAVA_POOL_SIZE_FOR_ESTIMATE	NUMBER	NOT NULL	Java pool size for the estimate (in megabytes)
JAVA_POOL_SIZE_FACTOR	NUMBER		Size factor with respect to the current Java pool size
ESTD_LC_SIZE	NUMBER		Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER		Estimated number of library cache memory objects in the Java pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER		Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a Java pool of the specified size. This is the time that would have been spent in reloading the required objects in the Java pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER		Estimated parse time saved factor with respect to the current Java pool size
ESTD_LC_LOAD_TIME	NUMBER		Estimated elapsed time (in seconds) for parsing in a Java pool of the specified size.
ESTD_LC_LOAD_TIME_FACTOR	NUMBER		Estimated load time factor with respect to the current Java pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER		Estimated number of times a library cache memory object was found in a Java pool of the specified size

DBA_HIST_LATCH

DBA_HIST_LATCH displays historical aggregate latch statistics for both parent and child latches, grouped by latch name. This view contains snapshots of V$LATCH.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch Hash
LATCH_NAME	VARCHAR2(64)		Latch name
LEVEL#	NUMBER		Latch level
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times

Column	Datatype	NULL	Description
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LATCH_CHILDREN

DBA_HIST_LATCH_CHILDREN displays historical statistics about child latches. This view includes all columns of DBA_HIST_LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other. This view contains snapshots of V$LATCH_CHILDREN.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch Hash
LATCH_NAME	VARCHAR2(64)		Latch name
CHILD#	NUMBER		Child latch number (unique only to each parent latch)
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LATCH_MISSES_SUMMARY

DBA_HIST_LATCH_MISSES_SUMMARY displays historical summary statistics about missed attempts to acquire a latch. This view contains snapshots of V$LATCH_MISSES.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
PARENT_NAME	VARCHAR2(50)		Latch name of a parent latch
WHERE_IN_CODE	VARCHAR2(64)		Location that attempted to acquire the latch
NWFAIL_COUNT	NUMBER		Number of times that no-wait acquisition of the latch failed

Column	Datatype	NULL	Description
SLEEP_COUNT	NUMBER		Number of times that acquisition attempts caused sleeps
WTR_SLP_COUNT	NUMBER		Number of times a waiter slept

DBA_HIST_LATCH_NAME

DBA_HIST_LATCH_NAME displays information about decoded latch names for the latches shown in DBA_HIST_LATCH. This view contains a snapshot of V$LATCHNAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
LATCH_HASH	NUMBER	NOT NULL	Latch hash
LATCH_NAME	VARCHAR2(64)	NOT NULL	Latch name

DBA_HIST_LATCH_PARENT

DBA_HIST_LATCH_PARENT displays historical statistics about parent latches. This view contains snapshots of V$LATCH_PARENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
LATCH_HASH	NUMBER		Latch hash
LATCH_NAME	VARCHAR2(64)		Latch name
LEVEL#	NUMBER		Latch level
GETS	NUMBER		Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER		Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER		Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER		Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER		Number of times a no-wait latch request did not succeed (that is, missed)
SPIN_GETS	NUMBER		Number of willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP1	NUMBER		Waits that slept 1 time
SLEEP2	NUMBER		Waits that slept 2 times
SLEEP3	NUMBER		Waits that slept 3 times
SLEEP4	NUMBER		Waits that slept 4 or more times
WAIT_TIME	NUMBER		Elapsed time spent waiting for the latch (in microseconds)

DBA_HIST_LIBRARYCACHE

DBA_HIST_LIBRARYCACHE displays historical statistics about library cache performance and activity. This view contains snapshots of V$LIBRARYCACHE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAMESPACE	VARCHAR2(15)	NOT NULL	Library cache namespace
GETS	NUMBER		Number of times a lock was requested for objects of the namespace
GETHITS	NUMBER		Number of times an object's handle was found in memory
PINS	NUMBER		Number of times a PIN was requested for objects of the namespace
PINHITS	NUMBER		Number of times all of the metadata pieces of the library object were found in memory
RELOADS	NUMBER		Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from disk
INVALIDATIONS	NUMBER		Total number of times objects in the namespace were marked invalid because a dependent object was modified
DLM_LOCK_REQUESTS	NUMBER		Number of GET requests lock instance locks
DLM_PIN_REQUESTS	NUMBER		Number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER		Number of release requests PIN instance locks
DLM_INVALIDATION_REQUESTS	NUMBER		Number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER		Number of invalidation pings received from other instances

DBA_HIST_LOG

DBA_HIST_LOG displays historical log file information from the control file. This view contains snapshots of V$LOG.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
GROUP#	NUMBER	NOT NULL	Log group number
THREAD#	NUMBER	NOT NULL	Log thread number
SEQUENCE#	NUMBER	NOT NULL	Log sequence number
BYTES	NUMBER		Size of the log (in bytes)
MEMBERS	NUMBER		Number of members in the log group
ARCHIVED	VARCHAR2(3)		Archive status (YES) or NO)

Column	Datatype	NULL	Description
STATUS	VARCHAR2(16)		Log status: ■ UNUSED - Online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. ■ CURRENT - Current redo log. This implies that the redo log is active. The redo log could be open or closed. ■ ACTIVE - Log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It may or may not be archived. ■ CLEARING - Log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE statement. After the log is cleared, the status changes to UNUSED. ■ CLEARING_CURRENT - Current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. ■ INACTIVE - Log is no longer needed for instance recovery. It may be in use for media recovery. It may or may not be archived. ■ INVALIDATED - Archived the current redo log without a log switch.
FIRST_CHANGE#	NUMBER		Lowest system change number (SCN) in the log
FIRST_TIME	DATE		Time of the first SCN in the log

DBA_HIST_MEM_DYNAMIC_COMP

DBA_HIST_MEM_DYNAMIC_COMP displays historical memory component sizes. This view contains snapshots of V$MEMORY_DYNAMIC_COMPONENTS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
COMPONENT	VARCHAR2(64)	NOT NULL	Component name
CURRENT_SIZE	NUMBER		Current size of the component
MIN_SIZE	NUMBER		Minimum size of the component since instance startup
MAX_SIZE	NUMBER		Maximum size of the component since instance startup
USER_SPECIFIED_SIZE	NUMBER		Value of the user parameter for the component
OPER_COUNT	NUMBER		Number of operations since instance startup
LAST_OPER_TYPE	VARCHAR2(13)		Last completed operation for the component: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL

Column	Datatype	NULL	Description
LAST_OPER_MODE	VARCHAR2(9)		Mode of the last completed operation: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
LAST_OPER_TIME	DATE		Start time of the last completed operation
GRANULE_SIZE	NUMBER		Granularity of the GROW or SHRINK operation

DBA_HIST_MEMORY_RESIZE_OPS

DBA_HIST_MEMORY_RESIZE_OPS displays memory resize operations history. This view contains snapshots of V$MEMORY_RESIZE_OPS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
COMPONENT	VARCHAR2(64)	NOT NULL	Component name
OPER_TYPE	VARCHAR2(13)	NOT NULL	Operation type: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL
START_TIME	DATE	NOT NULL	Start time of the operation
END_TIME	DATE	NOT NULL	End time of the operation
TARGET_SIZE	NUMBER	NOT NULL	Requested value of the parameter after the resize
OPER_MODE	VARCHAR2(9)		Operation mode: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
PARAMETER	VARCHAR2(80)		Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER		Parameter value at the start of the operation
FINAL_SIZE	NUMBER		Real value of the parameter after the resize
STATUS	VARCHAR2(9)		Completion status of the operation: ■ INACTIVE ■ PENDING ■ COMPLETE ■ CANCELLED ■ ERROR

DBA_HIST_MEMORY_TARGET_ADVICE

DBA_HIST_MEMORY_TARGET_ADVICE displays memory target advice history. This view contains snapshots of V$MEMORY_TARGET_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
MEMORY_SIZE	NUMBER	NOT NULL	If the MEMORY_SIZE_FACTOR column has a value of 1, then this column shows the current size of memory, as set by the MEMORY_TARGET initialization parameter. If the value of the MEMORY_SIZE_FACTOR column is less than or greater than 1, then this column shows a proposed memory size.
MEMORY_SIZE_FACTOR	NUMBER		A multiplier for the current memory size. Possible values are 0.25, 0.5, 0.75, 1, 1.5, 1.75 and 2. This multiplier times the current memory size equals the value of the MEMORY_SIZE column.
ESTD_DB_TIME	NUMBER		For current memory size (MEMORY_SIZE_FACTOR = 1), the amount of database time required to complete the current workload. For a proposed memory size, the estimated amount of database time that would be required if the MEMORY_TARGET parameter were changed to the proposed size.
ESTD_DB_TIME_FACTOR	NUMBER		For a proposed memory size, ratio of estimated database time to current database time
VERSION	NUMBER		Version number of this recommendation

DBA_HIST_METRIC_NAME

DBA_HIST_METRIC_NAME describes attributes of the set of RDBMS metrics. This view contains a snapshot of V$METRICNAME.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
GROUP_ID	NUMBER	NOT NULL	Metric Group ID
GROUP_NAME	VARCHAR2(64)		Metric group name
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement

DBA_HIST_MTTR_TARGET_ADVICE

DBA_HIST_MTTR_TARGET_ADVICE displays historical predictions of the number of physical I/O requests for the MTTR corresponding to each row. The data also includes a physical I/O factor, which is the ratio of the number of estimated I/O requests to the number of I/O requests actually performed by the current MTTR setting during the measurement interval. This view contains snapshots of V$MTTR_TARGET_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
MTTR_TARGET_FOR_ ESTIMATE	NUMBER		MTTR setting being simulated (equal to the current MTTR setting if this is the first row of the view)

Column	Datatype	NULL	Description
ADVICE_STATUS	VARCHAR2(5)		Current status of MTTR simulation: - ON SET - READY SET - OFF
DIRTY_LIMIT	NUMBER		Dirty buffer limit derived from the MTTR being simulated
ESTD_CACHE_WRITES	NUMBER		Estimated number of cache physical writes under the MTTR
ESTD_CACHE_WRITE_FACTOR	NUMBER		Estimated cache physical write ratio under the MTTR. It is the ratio of the estimated number of cache writes to the number of cache writes under the current MTTR setting.
ESTD_TOTAL_WRITES	NUMBER		Estimated total number of physical writes under the MTTR
ESTD_TOTAL_WRITE_FACTOR	NUMBER		Estimated total physical write ratio under the MTTR. It is the ratio of the estimated total number of physical writes to the total number of physical writes under the current MTTR setting.
ESTD_TOTAL_IOS	NUMBER		Estimated total number of I/O requests under the MTTR
ESTD_TOTAL_IO_FACTOR	NUMBER		Estimated total I/O ratio under the MTTR. It is the ratio of the estimated total number of I/O requests to the total number of I/O requests under the current MTTR setting.

DBA_HIST_MUTEX_SLEEP

DBA_HIST_MUTEX_SLEEP displays mutex sleep summary historical statistics information.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Databse identifier for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
MUTEX_TYPE	VARCHAR2(32)	NOT NULL	Mutex type
LOCATION	VARCHAR2(40)	NOT NULL	???
SLEEPS	NUMBER		???
WAIT_TIME	NUMBER		???

DBA_HIST_OPTIMIZER_ENV

DBA_HIST_OPTIMIZER_ENV displays the optimizer environments that have been captured in the Workload Repository. This view is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
OPTIMIZER_ENV_HASH_VALUE	NUMBER	NOT NULL	Hash value for the optimizer environment
OPTIMIZER_ENV	RAW(623)		Optimizer environment

DBA_HIST_OSSTAT

DBA_HIST_OSSTAT displays historical operating system statistics. This view contains snapshots of V$OSSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic ID
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_OSSTAT_NAME

DBA_HIST_OSSTAT_NAME displays the names of the operating system statistics. This view is used with DBA_HIST_OSSTAT.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
STAT_ID	NUMBER	NOT NULL	Statistic ID
STAT_NAME	VARCHAR2(64)	NOT NULL	Statistic name

DBA_HIST_PARAMETER

DBA_HIST_PARAMETER displays historical information about the initialization parameters that were in effect for the instance. This view contains snapshots of V$SYSTEM_PARAMETER.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
PARAMETER_HASH	NUMBER		Parameter hash
PARAMETER_NAME	VARCHAR2(64)		Name of the parameter
VALUE	VARCHAR2(512)		Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value
ISDEFAULT	VARCHAR2(9)		Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISMODIFIED	VARCHAR2(10)		Indicates whether the parameter has been modified after instance startup: ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup

DBA_HIST_PARAMETER_NAME

DBA_HIST_PARAMETER_NAME displays information about the parameter names captured in the workload repository. This view is used with the DBA_HIST_PARAMETER view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
PARAMETER_HASH	NUMBER	NOT NULL	Parameter hash
PARAMETER_NAME	VARCHAR2(64)	NOT NULL	Name of the parameter

DBA_HIST_PERSISTENT_QUEUES

DBA_HIST_PERSISTENT_QUEUES displays STREAMS AQ persistent queues historical statistics information. This view contains snapshots of V$PERSISTENT_QUEUES.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
QUEUE_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
QUEUE_ID	NUMBER	NOT NULL	Identifier for the queue
FIRST_ACTIVITY_TIME	TIMESTAMP(6)		First queue activity time since database startup
ENQUEUED_MSGS	NUMBER		Number of messages enqueued
DEQUEUED_MSGS	NUMBER		Number of messages dequeued **Note:** This column will not be incremented until all the subscribers of the message have dequeued the message and its retention time has elapsed.
ELAPSED_ENQUEUE_TIME	NUMBER		Total time (in hundredths of a second) spent doing enqueue
ELAPSED_DEQUEUE_TIME	NUMBER		Total time (in hundredths of a second) spent doing dequeue
ELAPSED_TRANSFORMATION_TIME	NUMBER		Total time (in hundredths of a second) spent doing transformation
ELAPSED_RULE_EVALUATION_TIME	NUMBER		Total time (in hundredths of a second) spent doing rule evaluation
ENQUEUED_EXPIRY_MSGS	NUMBER		Number of messages enqueued with expiry
ENQUEUED_DELAY_MSGS	NUMBER		Number of messages enqueued with delay
MSGS_MADE_EXPIRED	NUMBER		Number of messages expired by time manager
MSGS_MADE_READY	NUMBER		Number of messages made ready by time manager
LAST_ENQUEUE_TIME	TIMESTAMP(6)		Last message enqueue time
LAST_DEQUEUE_TIME	TIMESTAMP(6)		Last message dequeue time
LAST_TM_EXPIRY_TIME	TIMESTAMP(6)		Last time message was expired by time manager
LAST_TM_READY_TIME	TIMESTAMP(6)		Last time message was made ready by time manager

DBA_HIST_PERSISTENT_SUBS

DBA_HIST_PERSISTENT_SUBS displays STREAMS AQ persistent queue subscribers historical statistics information. This view contains snapshots of V$PERSISTENT_SUBSCRIBERS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
QUEUE_SCHEMA	VARCHAR2(30)	NOT NULL	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	NOT NULL	Name of the queue
SUBSCRIBER_ID	NUMBER	NOT NULL	Internal subscriber number
SUBSCRIBER_NAME	VARCHAR2(30)		Name of the subscriber
SUBSCRIBER_ADDRESS	VARCHAR2(1024)		Address of the subscribing agent
SUBSCRIBER_TYPE	VARCHAR2(30)		Type of the subscriber: ▪ PROXY - Propagation subscriber ▪ SUBSCRIBER - Normal subscriber ▪ RECIPIENT - Recipient
FIRST_ACTIVITY_TIME	TIMESTAMP(6)		First subscriber activity time since database startup
ENQUEUED_MSGS	NUMBER		Number of messages enqueued since FIRST_ACTIVITY_TIME
DEQUEUED_MSGS	NUMBER		Number of messages dequeued since FIRST_ACTIVITY_TIME
EXPIRED_MSGS	NUMBER		Number of messages expired since FIRST_ACTIVITY_TIME
DEQUEUED_MSG_LATENCY	NUMBER		Last dequeued message latency (in seconds)
LAST_ENQUEUE_TIME	TIMESTAMP(6)		Timestamp of the last enqueued message
LAST_DEQUEUE_TIME	TIMESTAMP(6)		Timestamp of the last dequeued message

DBA_HIST_PGA_TARGET_ADVICE

DBA_HIST_PGA_TARGET_ADVICE displays historical predictions of how the cache hit percentage and over allocation count statistics displayed by the V$PGASTAT performance view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. This view contains snapshots of V$PGA_TARGET_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
PGA_TARGET_FOR_ESTIMATE	NUMBER	NOT NULL	Value of PGA_AGGREGATE_TARGET for the prediction (in bytes)
PGA_TARGET_FACTOR	NUMBER		PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2(3)		Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
BYTES_PROCESSED	NUMBER		Total bytes processed by all the work areas considered by this advice (in bytes)

Column	Datatype	NULL	Description
ESTD_TIME	NUMBER		???
ESTD_EXTRA_BYTES_RW	NUMBER		Estimated number of extra bytes which would be read or written if PGA_AGGREGATE_TARGET was set to the value of the PGA_TARGET_FOR_ESTIMATE column. This number is derived from the estimated number and size of work areas which would run in one-pass (or multi-pass) for that value of PGA_AGGREGATE_TARGET.
ESTD_PGA_CACHE_HIT_PERCENTAGE	NUMBER		Estimated value of the cache hit percentage statistic when PGA_AGGREGATE_TARGET equals PGA_TARGET_FOR_ESTIMATE. This column is derived from the above two columns and is equal to BYTES_PROCESSED / (BYTES_PROCESSED + ESTD_EXTRA_BYTES_RW)
ESTD_OVERALLOC_COUNT	NUMBER		Estimated number of PGA memory over-allocations if the value of PGA_AGGREGATE_TARGET is set to PGA_TARGET_FOR_ESTIMATE. A nonzero value means that PGA_TARGET_FOR_ESTIMATE is not large enough to run the work area workload. Hence, PGA_AGGREGATE_TARGET should not be set to PGA_TARGET_FOR_ESTIMATE since Oracle will not be able to honor that target.

DBA_HIST_PGASTAT

DBA_HIST_PGASTAT displays historical PGA memory usage statistics as well as statistics about the automatic PGA memory manager when it is enabled. This view contains snapshots of V$PGASTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Unique snapshot ID
NAME	VARCHAR2(64)	NOT NULL	Name of the statistic: ■ aggregate PGA target parameter ■ aggregate PGA auto target ■ global memory bound ■ total PGA allocated ■ total PGA used ■ total PGA used for auto workareas ■ total PGA used for manual workareas ■ over allocation count ■ bytes processed ■ extra bytes read/written ■ cache hit percentage **See Also:** V$PGASTAT for descriptions of the statistics
VALUE	NUMBER		Statistic value

DBA_HIST_PROCESS_MEM_SUMMARY

DBA_HIST_PROCESS_MEM_SUMMARY displays historical information about dynamic PGA memory usage by named component categories for each process.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
CATEGORY	VARCHAR2(15)	NOT NULL	Category name. Categories include "SQL", "PL/SQL", "OLAP" and "JAVA". Special categories are "Freeable" and "Other". Freeable memory has been allocated to the process by the operating system, but has not been allocated to a category. "Other" memory has been allocated to a category, but not to one of the named categories
NUM_PROCESSES	NUMBER		Number of processes
NON_ZERO_ALLOCS	NUMBER		Number of processes with non-zero allocations
USED_TOTAL	NUMBER		Bytes of PGA memory used by the process for the category
ALLOCATED_TOTAL	NUMBER		Total number of bytes of PGA memory allocated by the process for the category.
ALLOCATED_AVG	NUMBER		Average number of bytes of PGA memory allocated by the process for the category
ALLOCATED_STDDEV	NUMBER		Standard deviation of the number of bytes of PGA memory allocated by the process for the category
ALLOCATED_MAX	NUMBER		Maximum bytes of PGA memory ever allocated by the process for the category
MAX_ALLOCATED_MAX	NUMBER		Maximum bytes of PGA memory that can be allocated by the process for the category

DBA_HIST_RESOURCE_LIMIT

DBA_HIST_RESOURCE_LIMIT displays historical information about global resource use for some of the system resource. This view contains snapshots of V$RESOURCE_LIMIT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
RESOURCE_NAME	VARCHAR2(30)	NOT NULL	Name of the resource
CURRENT_UTILIZATION	NUMBER		Number of (resources, locks, or processes) currently being used
MAX_UTILIZATION	NUMBER		Maximum consumption of the resource since the last instance start up
INITIAL_ALLOCATION	VARCHAR2(10)		Initial allocation. This will be equal to the value specified for the resource in the initialization parameter file (UNLIMITED for infinite allocation).
LIMIT_VALUE	VARCHAR2(10)		Unlimited for resources and locks. This can be greater than the initial allocation value (UNLIMITED for infinite limit).

DBA_HIST_ROWCACHE_SUMMARY

DBA_HIST_ROWCACHE_SUMMARY displays historical summary statistics for data dictionary activity. This view contains snapshots of V$ROWCACHE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
PARAMETER	VARCHAR2(32)		Name of the initialization parameter that determines the number of entries in the data dictionary cache
TOTAL_USAGE	NUMBER		Sum of the total number of entries in the cache
USAGE	NUMBER		Number of cache entries that contain valid data
GETS	NUMBER		Total number of requests for information on the data object
GETMISSES	NUMBER		Number of data requests resulting in cache misses
SCANS	NUMBER		Number of scan requests
SCANMISSES	NUMBER		Number of times a scan failed to find the data in the cache
SCANCOMPLETES	NUMBER		For a list of subordinate entries, the number of times the list was scanned completely
MODIFICATIONS	NUMBER		Number of inserts, updates, and deletions
FLUSHES	NUMBER		Number of times flushed to disk
DLM_REQUESTS	NUMBER		Number of DLM requests
DLM_CONFLICTS	NUMBER		Number of DLM conflicts
DLM_RELEASES	NUMBER		Number of DLM releases

DBA_HIST_RSRC_CONSUMER_GROUP

DBA_HIST_RSRC_CONSUMER_GROUP displays historical information about Resource Manager consumer groups. This view contains snapshots of V$RSRC_CONS_GROUP_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SEQUENCE#	NUMBER	NOT NULL	A sequential counter that uniquely describes the DBA_HIST_RSRC_PLAN entry to which these consumer group statistics apply. When the instance is restarted, this value is reset to zero.
CONSUMER_GROUP_ID	NUMBER	NOT NULL	Consumer group object ID (a unique number, consistent across database shutdowns and startups)
CONSUMER_GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the consumer group
REQUESTS	NUMBER	NOT NULL	Cumulative number of requests that were executed in the consumer group
CPU_WAIT_TIME	NUMBER	NOT NULL	Cumulative amount of time that sessions waited for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CPU_WAITS	NUMBER	NOT NULL	Cumulative number of times all sessions in the consumer group had to wait for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CONSUMED_CPU_TIME	NUMBER	NOT NULL	Cumulative amount of CPU time consumed by all sessions in the consumer group

Column	Datatype	NULL	Description
YIELDS	NUMBER	NOT NULL	Cumulative number of times that sessions in the consumer group had to yield CPU to other sessions because of quantum expiration
ACTIVE_SESS_LIMIT_HIT	NUMBER	NOT NULL	Number of times that sessions in the consumer group were queued because the consumer group reached its active session limit
UNDO_LIMIT_HIT	NUMBER	NOT NULL	Number of times that queries in the consumer group were cancelled because the consumer group reached its UNDO_POOL limit
SWITCHES_IN_CPU_TIME	NUMBER	NOT NULL	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_OUT_CPU_TIME	NUMBER	NOT NULL	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_IN_IO_MEGABYTES	NUMBER	NOT NULL	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_OUT_IO_MEGABYTES	NUMBER	NOT NULL	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_IN_IO_REQUESTS	NUMBER	NOT NULL	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SWITCHES_OUT_IO_REQUESTS	NUMBER	NOT NULL	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SQL_CANCELED	NUMBER	NOT NULL	Number of times that SQL queries running in the consumer group were aborted because they exceeded the Resource Manager plan's SWITCH_TIME limit and CANCEL_SQL was specified as the Resource Manager plan's SWITCH_GROUP
ACTIVE_SESS_KILLED	NUMBER	NOT NULL	Number of times that sessions running in the consumer group were terminated because they exceeded the Resource Manager plan's SWITCH_TIME limit and KILL_SESSION was specified as the Resource Manager plan's SWITCH_GROUP
IDLE_SESS_KILLED	NUMBER	NOT NULL	Number of times that sessions in the consumer group were killed because they were idle for too long (reached MAX_IDLE_TIME)
IDLE_BLKR_SESS_KILLED	NUMBER	NOT NULL	Number of times that sessions in the consumer group were killed because they were idle too long (reached MAX_IDLE_BLOCKER_TIME) and were blocking other sessions
QUEUED_TIME	NUMBER	NOT NULL	Total amount of time that sessions in the consumer group have spent in the QUEUED state because of the active session limit
QUEUE_TIME_OUTS	NUMBER	NOT NULL	Number of times that requests from sessions in the consumer group timed out because they were queued for too long (reached QUEUEING_P1)
IO_SERVICE_TIME	NUMBER	NOT NULL	Cumulative I/O wait time (in microseconds)
IO_SERVICE_WAITS	NUMBER	NOT NULL	Total number of wait requests
SMALL_READ_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	NOT NULL	Number of multiblock megabytes written
SMALL_READ_REQUESTS	NUMBER	NOT NULL	Number of single block read requests
SMALL_WRITE_REQUESTS	NUMBER	NOT NULL	Number of single block write requests
LARGE_READ_REQUESTS	NUMBER	NOT NULL	Number of multiblock read requests

Column	Datatype	NULL	Description
LARGE_WRITE_REQUESTS	NUMBER	NOT NULL	Number of multiblock write requests

DBA_HIST_RSRC_PLAN

DBA_HIST_RSRC_PLAN displays historical information about resource plans. This view contains snapshots of V$RSRC_PLAN_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SEQUENCE#	NUMBER	NOT NULL	A sequential counter that uniquely describes a row. When the instance is restarted, this value is reset to zero.
START_TIME	DATE	NOT NULL	Time that the resource plan was enabled
END_TIME	DATE	NOT NULL	Time that the resource plan was disabled; NULL if the row contains the current resource plan information
PLAN_ID	NUMBER	NOT NULL	Resource plan ID; NULL if the Resource Manager was disabled
PLAN_NAME	VARCHAR2(30)	NOT NULL	Resource plan name; NULL if the Resource Manager was disabled
CPU_MANAGED	VARCHAR2(4)	NOT NULL	Indicates whether the resource plan has parameters that specify a policy for how the Resource Manager should schedule sessions to manage CPU usage (ON) or whether Resource Manager is not managing CPU usage (OFF)

DBA_HIST_RULE_SET

DBA_HIST_RULE_SET displays historical information about rule set statistics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
OWNER	VARCHAR2(30)	NOT NULL	Owner of the rule set
NAME	VARCHAR2(30)	NOT NULL	Name of the rule set
STARTUP_TIME	DATE	NOT NULL	Startup time of the instance
CPU_TIME	NUMBER		Total CPU time (in hundredths of a second) spent in evaluation of the rule set
ELAPSED_TIME	NUMBER		Total elapsed time (in hundredths of a second) spent in evaluation of the rule set
EVALUATIONS	NUMBER		Number of evaluations on the rule set
SQL_FREE_EVALUATIONS	NUMBER		Number of evaluations on the rule set which did not internally issue SQL to evaluate rules
SQL_EXECUTIONS	NUMBER		Total number of SQL statements executed during evaluation of the rule set
RELOADS	NUMBER		Number of times the rule set object was reloaded in shared memory

DBA_HIST_SEG_STAT

DBA_HIST_SEG_STAT displays historical information about segment-level statistics. This view captures the top segments based on a set of criteria and captures information from V$SEGSTAT. The total value is the value of the statistics since instance startup. The delta value is the value of the statistics from the BEGIN_INTERVAL_TIME to the END_INTERVAL_TIME in the DBA_HIST_SNAPSHOT view.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
TS#	NUMBER		Tablespace number
OBJ#	NUMBER		Dictionary object number
DATAOBJ#	NUMBER		Data object number
LOGICAL_READS_TOTAL	NUMBER		Cumulative value for logical reads
LOGICAL_READS_DELTA	NUMBER		Delta value for logical reads
BUFFER_BUSY_WAITS_TOTAL	NUMBER		Cumulative value for buffer busy waits
BUFFER_BUSY_WAITS_DELTA	NUMBER		Delta value for buffer busy waits
DB_BLOCK_CHANGES_TOTAL	NUMBER		Cumulative value for db block changes
DB_BLOCK_CHANGES_DELTA	NUMBER		Delta value for db block changes
PHYSICAL_READS_TOTAL	NUMBER		Cumulative value for physical reads
PHYSICAL_READS_DELTA	NUMBER		Delta value for physical reads
PHYSICAL_WRITES_TOTAL	NUMBER		Cumulative value for physical writes
PHYSICAL_WRITES_DELTA	NUMBER		Delta value for physical writes
PHYSICAL_READS_DIRECT_TOTAL	NUMBER		Cumulative value for physical reads direct
PHYSICAL_READS_DIRECT_DELTA	NUMBER		Delta value for physical reads direct
PHYSICAL_WRITES_DIRECT_TOTAL	NUMBER		Cumulative value for physical writes direct
PHYSICAL_WRITES_DIRECT_DELTA	NUMBER		Delta value for physical writes direct
ITL_WAITS_TOTAL	NUMBER		Cumulative value for ITL waits
ITL_WAITS_DELTA	NUMBER		Delta value for ITL waits
ROW_LOCK_WAITS_TOTAL	NUMBER		Cumulative value for row lock waits
ROW_LOCK_WAITS_DELTA	NUMBER		Delta value for row lock waits
GC_CR_BLOCKS_SERVED_TOTAL	NUMBER		Cumulative value for global cache CR blocks served
GC_CR_BLOCKS_SERVED_DELTA	NUMBER		Delta value for global cache CR blocks served
GC_CU_BLOCKS_SERVED_TOTAL	NUMBER		Cumulative value for global cache current blocks served
GC_CU_BLOCKS_SERVED_DELTA	NUMBER		Delta value for global cache current blocks served
GC_BUFFER_BUSY_TOTAL	NUMBER		Cumulative value for global cache buffer busy
GC_BUFFER_BUSY_DELTA	NUMBER		Delta value for global cache buffer busy

Column	Datatype	NULL	Description
GC_CR_BLOCKS_ RECEIVED_TOTAL	NUMBER		Cumulative value for global cache CR blocks received
GC_CR_BLOCKS_ RECEIVED_DELTA	NUMBER		Delta value for global cache CR blocks received
GC_CU_BLOCKS_ RECEIVED_TOTAL	NUMBER		Cumulative value for global cache current blocks received
GC_CU_BLOCKS_ RECEIVED_DELTA	NUMBER		Delta value for global cache current blocks received
SPACE_USED_TOTAL	NUMBER		Cumulative value for space used
SPACE_USED_DELTA	NUMBER		Delta value for space used
SPACE_ALLOCATED_TOTAL	NUMBER		Cumulative value for space allocated
SPACE_ALLOCATED_DELTA	NUMBER		Delta value for space allocated
TABLE_SCANS_TOTAL	NUMBER		Cumulative value for table scans
TABLE_SCANS_DELTA	NUMBER		Delta value for table scans

DBA_HIST_SEG_STAT_OBJ

DBA_HIST_SEG_STAT_OBJ displays all the names of the segments captured in the workload repository. This view is used with the DBA_HIST_SEG_STAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
TS#	NUMBER	NOT NULL	Tablespace number
OBJ#	NUMBER	NOT NULL	Dictionary object number
DATAOBJ#	NUMBER	NOT NULL	Data object number
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(18)		Type of the object
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace Name for the object
PARTITION_TYPE	VARCHAR2(8)		Partition Type, if relevant

DBA_HIST_SERVICE_NAME

DBA_HIST_SERVICE_NAME displays the names of the Services tracked by the Workload Repository. This view contains information for V$SERVICES.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
SERVICE_NAME_HASH	NUMBER	NOT NULL	Hash of the service name
SERVICE_NAME	VARCHAR2(64)	NOT NULL	Name of the service

DBA_HIST_SERVICE_STAT

DBA_HIST_SERVICE_STAT displays the history of important service statistics tracked by the Workload Repository. The call rate statistics in this view can be used for making run-time routing decisions, for tracking service levels, and for per-instance diagnostics per call rate.

The elapsed timing for each call provides a relative value across instances for how well a node is processing SQL calls issued under a service name. When aggregation is enabled for the service name, this view provides historical data on the timing and work done for calls issued for the whole service. This view contains information from V$SERVICE_STATS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SERVICE_NAME_HASH	NUMBER		Hash of the service name
SERVICE_NAME	VARCHAR2(64)		Name of the service
STAT_ID	NUMBER		Statistic identifier
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Value of the statistic

DBA_HIST_SERVICE_WAIT_CLASS

DBA_HIST_SERVICE_WAIT_CLASS displays the history of wait class information for services as tracked by the Workload Repository. This view contains information from V$SERVICE_WAIT_CLASS.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SERVICE_NAME_HASH	NUMBER		Hash of the service name
SERVICE_NAME	VARCHAR2(64)		Name of the service
WAIT_CLASS_ID	NUMBER		Identifier for the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name for the class of the wait event
TOTAL_WAITS	NUMBER		Total number of waits for this event
TIME_WAITED	NUMBER		Total amount of time waited for this event (in hundredths of a second)

DBA_HIST_SESS_TIME_STATS

DBA_HIST_SESS_TIME_STATS displays information about CPU and I/O time for interesting Streams sessions.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SESSION_TYPE	VARCHAR2(64)	NOT NULL	Type of session
MIN_LOGON_TIME	DATE		Minimum logon time
SUM_CPU_TIME	NUMBER		Total CPU time
SUM_SYS_IO_WAIT	NUMBER		Total system I/O wait time
SUM_USER_IO_WAIT	NUMBER		Total user I/O wait time

DBA_HIST_SESSMETRIC_HISTORY

DBA_HIST_SESSMETRIC_HISTORY displays the history of several important session metrics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
SESSID	NUMBER	NOT NULL	Session ID
SERIAL#	NUMBER	NOT NULL	Session serial number
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
VALUE	NUMBER	NOT NULL	Metric Value
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement

DBA_HIST_SGA

DBA_HIST_SGA displays historical summary information about the system global area (SGA). This view contains snapshots of V$SGA.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
NAME	VARCHAR2(64)	NOT NULL	SGA component group
VALUE	NUMBER	NOT NULL	Memory size (in bytes)

DBA_HIST_SGA_TARGET_ADVICE

DBA_HIST_SGA_TARGET_ADVICE provides historical information about the SGA_TARGET initialization parameter.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SGA_SIZE	NUMBER	NOT NULL	Size of the SGA
SGA_SIZE_FACTOR	NUMBER	NOT NULL	Ratio between the SGA_SIZE and the current size of the SGA
ESTD_DB_TIME	NUMBER	NOT NULL	Estimated DB_TIME for this SGA_SIZE
ESTD_PHYSICAL_READS	NUMBER		Estimated number of physical reads

DBA_HIST_SGASTAT

DBA_HIST_SGASTAT displays detailed historical information on the system global area (SGA). This view contains snapshots of V$SGASTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
NAME	VARCHAR2(64)		SGA component group
POOL	VARCHAR2(12)		Designates the pool in which the memory in NAME resides: ▪ shared pool - Memory is allocated from the shared pool ▪ large pool - Memory is allocated from the large pool ▪ java pool - Memory is allocated from the Java pool ▪ streams pool - Memory is allocated from the Streams pool
BYTES	NUMBER		Memory size (in bytes)

DBA_HIST_SHARED_POOL_ADVICE

DBA_HIST_SHARED_POOL_ADVICE displays historical information about estimated parse time in the shared pool for different pool sizes. This view contains snapshots of V$SHARED_POOL_ADVICE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SHARED_POOL_SIZE_FOR_ESTIMATE	NUMBER	NOT NULL	Shared pool size for the estimate (in megabytes)
SHARED_POOL_SIZE_FACTOR	NUMBER		Size factor with respect to the current shared pool size
ESTD_LC_SIZE	NUMBER		Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER		Estimated number of library cache memory objects in the shared pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER		Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a shared pool of the specified size. This is the time that would have been spent in reloading the required objects in the shared pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER		Estimated parse time saved factor with respect to the current shared pool size
ESTD_LC_LOAD_TIME	NUMBER		Estimated elapsed time (in seconds) for parsing in a shared pool of the specified size.
ESTD_LC_LOAD_TIME_FACTOR	NUMBER		Estimated load time factor with respect to the current shared pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER		Estimated number of times a library cache memory object was found in a shared pool of the specified size

DBA_HIST_SNAP_ERROR

DBA_HIST_SNAP_ERROR displays information about the snapshot error information in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table in which the error occurred
ERROR_NUMBER	NUMBER	NOT NULL	Error number for the error encountered

DBA_HIST_SNAPSHOT

DBA_HIST_SNAPSHOT displays information about the snapshots in the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
STARTUP_TIME	TIMESTAMP(3)	NOT NULL	Startup time of the instance
BEGIN_INTERVAL_TIME	TIMESTAMP(3)	NOT NULL	Time at the beginning of the snapshot interval
END_INTERVAL_TIME	TIMESTAMP(3)	NOT NULL	Time at the end of the snapshot interval; the actual time the snapshot was taken
FLUSH_ELAPSED	INTERVAL DAY(5) TO SECOND(1)		Amount of time to perform the snapshot
SNAP_LEVEL	NUMBER		Snapshot level
ERROR_COUNT	NUMBER		Number of errors occurring in the tables for the particular snapshot

DBA_HIST_SQL_BIND_METADATA

DBA_HIST_SQL_BIND_METADATA displays historical information on metadata for bind variables used by SQL cursors.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID for the snapshot
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
NAME	VARCHAR2(30)		Name of the bind variable
POSITION	NUMBER	NOT NULL	Position of the bind variable in the SQL statement
DUP_POSITION	NUMBER		If the binding is performed by name and the bind variable is duplicated, then this column gives the position of the primary bind variable
DATATYPE	NUMBER		Internal identifier for the bind datatype
DATATYPE_STRING	VARCHAR2(15)		Textual representation of the bind datatype
CHARACTER_SID	NUMBER		National character set identifier
PRECISION	NUMBER		Precision (for numeric binds)
SCALE	NUMBER		Scale (for numeric binds)

Column	Datatype	NULL	Description
MAX_LENGTH	NUMBER		Maximum bind length

DBA_HIST_SQL_PLAN

DBA_HIST_SQL_PLAN displays the execution plan information for each child cursor in the workload repository. This view captures information from V$SQL_PLAN and is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
ID	NUMBER	NOT NULL	A number assigned to each step in the execution plan
OPERATION	VARCHAR2(30)		Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)		A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER		Object number of the table or the index
OBJECT_OWNER	VARCHAR2(30)		Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)		Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)		Alias for the object
OBJECT_TYPE	VARCHAR2(20)		Type of the object
OPTIMIZER	VARCHAR2(20)		Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
PARENT_ID	NUMBER		ID of the next execution step that operates on the output of the current step
DEPTH	NUMBER		Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER		Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER		Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER		Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER		Estimate, by the cost-based optimizer, of the number of bytes produced by the operation

Column	Datatype	NULL	Description
OTHER_TAG	VARCHAR2(35)		Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER		Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)		Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)		Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER		CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER		I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER		Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)		Expressions produced by the operation
TIME	NUMBER		Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)		Name of the query block
REMARKS	VARCHAR2(4000)		Remarks
TIMESTAMP	DATE		Timestamp for when the plan was produced
OTHER_XML	CLOB		Provides extra information specific to an execution step of the execution plan. The content of this column is structured using XML because it allows multiple pieces of information to be stored, including the following: ■ Name of the schema against which the query was parsed ■ Release number of the Oracle Database that produced the explain plan ■ Hash value associated with the execution plan ■ Name (if any) of the outline or the SQL profile used to build the execution plan ■ Indication of whether or not dynamic sampling was used to produce the plan ■ The outline data, a set of optimizer hints that can be used to regenerate the same plan

DBA_HIST_SQL_SUMMARY

DBA_HIST_SQL_SUMMARY displays historical SQL summary information.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot

Column	Datatype	NULL	Description
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
TOTAL_SQL	NUMBER	NOT NULL	Total number of SQL
TOTAL_SQL_MEM	NUMBER	NOT NULL	Total memory usage of SQL
SINGLE_USE_SQL	NUMBER	NOT NULL	Number of single use SQL
SINGLE_USE_SQL_MEM	NUMBER	NOT NULL	Total memory usage of single use SQL

DBA_HIST_SQL_WORKAREA_HSTGRM

DBA_HIST_SQL_WORKAREA_HSTGRM displays the historical cumulative work area execution statistics (cumulated since instance startup) for different work area groups. This view contains snapshots of V$SQL_WORKAREA_HISTOGRAM.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
LOW_OPTIMAL_SIZE	NUMBER	NOT NULL	Lower bound for the optimal memory requirement of work areas included in the row (in bytes)
HIGH_OPTIMAL_SIZE	NUMBER	NOT NULL	Upper bound for the optimal memory requirement of work areas included in the row (in bytes)
OPTIMAL_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in optimal mode since instance startup
ONEPASS_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in one-pass mode since instance startup
MULTIPASSES_EXECUTIONS	NUMBER		Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in multi-pass mode since instance startup
TOTAL_EXECUTIONS	NUMBER		Sum of OPTIMAL_EXECUTIONS, ONEPASS_EXECUTIONS, and MULTIPASSES_EXECUTIONS

DBA_HIST_SQLBIND

DBA_HIST_SQLBIND displays historical information on bind variables used by SQL cursors. This view contains snapshots of V$SQL_BIND_CAPTURE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SQL_ID	VARCHAR2(13)		SQL identifier of the parent cursor in the library cache
NAME	VARCHAR2(30)		Name of the bind variable
POSITION	NUMBER		Position of the bind variable in the SQL statement
DUP_POSITION	NUMBER		If the binding is performed by name and the bind variable is duplicated, then this column gives the position of the primary bind variable.
DATATYPE	NUMBER		Internal identifier for the bind datatype

Column	Datatype	NULL	Description
DATATYPE_STRING	VARCHAR2(15)		Textual representation of the bind datatype
CHARACTER_SID	NUMBER		National character set identifier
PRECISION	NUMBER		Precision (for numeric binds)
SCALE	NUMBER		Scale (for numeric binds)
MAX_LENGTH	NUMBER		Maximum bind length
WAS_CAPTURED	VARCHAR2(3)		Indicates whether the bind value was captured (YES) or not (NO)
LAST_CAPTURED	DATE		Date when the bind value was captured. Bind values are captured when SQL statements are executed. To limit the overhead, binds are captured at most every 15 minutes for a given cursor.
VALUE_STRING	VARCHAR2(4000)		Value of the bind represented as a string
VALUE_ANYDATA	ANYDATA		Value of the bind represented using the self-descriptive Sys.AnyData datatype. This representation is useful to programmatically decode the value of the bind variable.

DBA_HIST_SQLSTAT

DBA_HIST_SQLSTAT displays historical information about SQL statistics. This view captures the top SQL statements based on a set of criteria and captures the statistics information from V$SQL. The total value is the value of the statistics since instance startup. The delta value is the value of the statistics from the BEGIN_INTERVAL_TIME to the END_INTERVAL_TIME in the DBA_HIST_SNAPSHOT view.

This view is used with the DBA_HIST_OPTIMIZER_ENV, DBA_HIST_SQLTEXT, and DBA_HIST_SQL_PLAN views to provide a complete picture of historical SQL statistics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
SQL_ID	VARCHAR2(13)		SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER		Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
OPTIMIZER_COST	NUMBER		Cost of the query given by the optimizer
OPTIMIZER_MODE	VARCHAR2(10)		Mode under which the SQL statement is executed
OPTIMIZER_ENV_HASH_VALUE	NUMBER		Hash Value for the optimizer environment
SHARABLE_MEM	NUMBER		Amount of shared memory used by the child cursor (in bytes)
LOADED_VERSIONS	NUMBER		Indicates whether the context heap is loaded (1) or not (0)
VERSION_COUNT	NUMBER		Number of children associated with the cursor
MODULE	VARCHAR2(64)		Contains the name of the module that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_MODULE
ACTION	VARCHAR2(64)		Contains the name of the action that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_ACTION

Column	Datatype	NULL	Description
SQL_PROFILE	VARCHAR2(64)		Name of the applied SQL Profile
FORCE_MATCHING_ SIGNATURE	NUMBER		The signature used when the CURSOR_SHARING parameter is set to FORCE
PARSING_SCHEMA_ID	NUMBER		Schema ID that was used to originally build the child cursor
PARSING_SCHEMA_NAME	VARCHAR2(30)		Schema name that was used to originally build the child cursor
FETCHES_TOTAL	NUMBER		Cumulative number of fetches associated with the SQL statement
FETCHES_DELTA	NUMBER		Delta number of fetches associated with the SQL statement
END_OF_FETCH_COUNT_ TOTAL	NUMBER		Cumulative number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
END_OF_FETCH_COUNT_ DELTA	NUMBER		Delta number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed.
SORTS_TOTAL	NUMBER		Cumulative number of sorts that were done for this child cursor
SORTS_DELTA	NUMBER		Delta number of sorts that were done for this child cursor
EXECUTIONS_TOTAL	NUMBER		Cumulative number of executions that took place on this object since it was brought into the library cache
EXECUTIONS_DELTA	NUMBER		Delta number of executions that took place on this object since it was brought into the library cache
PX_SERVERS_EXECS_ TOTAL	NUMBER		Cumulative number of PX server executions
PX_SERVERS_EXECS_ DELTA	NUMBER		Delta number of PX server executions
LOADS_TOTAL	NUMBER		Cumulative number of times the object was either loaded or reloaded
LOADS_DELTA	NUMBER		Delta number of times the object was either loaded or reloaded
INVALIDATIONS_TOTAL	NUMBER		Cumulative number of times this child cursor has been invalidated
INVALIDATIONS_DELTA	NUMBER		Delta number of times this child cursor has been invalidated
PARSE_CALLS_TOTAL	NUMBER		Cumulative number of parse calls for this child cursor
PARSE_CALLS_DELTA	NUMBER		Delta number of parse calls for this child cursor
DISK_READS_TOTAL	NUMBER		Cumulative number of disk reads for this child cursor
DISK_READS_DELTA	NUMBER		Delta number of disk reads for this child cursor
BUFFER_GETS_TOTAL	NUMBER		Cumulative number of buffer gets for this child cursor
BUFFER_GETS_DELTA	NUMBER		Delta number of buffer gets for this child cursor
ROWS_PROCESSED_TOTAL	NUMBER		Cumulative number of rows the parsed SQL statement returns

Column	Datatype	NULL	Description
ROWS_PROCESSED_DELTA	NUMBER		Delta number of rows the parsed SQL statement returns
CPU_TIME_TOTAL	NUMBER		Cumulative value of CPU time (in microseconds) used by this cursor for parsing/executing/fetching
CPU_TIME_DELTA	NUMBER		Delta value of CPU time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME_TOTAL	NUMBER		Cumulative value of elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
ELAPSED_TIME_DELTA	NUMBER		Delta value of elapsed time (in microseconds) used by this cursor for parsing/executing/fetching
IOWAIT_TOTAL	NUMBER		Cumulative value of user I/O wait time (in microseconds)
IOWAIT_DELTA	NUMBER		Delta value of user I/O wait time (in microseconds)
CLWAIT_TOTAL	NUMBER		Cumulative value of cluster wait time (in microseconds)
CLWAIT_DELTA	NUMBER		Delta value of cluster wait time (in microseconds)
APWAIT_TOTAL	NUMBER		Cumulative value of application wait time (in microseconds)
APWAIT_DELTA	NUMBER		Delta value of application wait time (in microseconds)
CCWAIT_TOTAL	NUMBER		Cumulative value of concurrency wait time (in microseconds)
CCWAIT_DELTA	NUMBER		Delta value of concurrency wait time (in microseconds)
DIRECT_WRITES_TOTAL	NUMBER		Cumulative value of direct writes
DIRECT_WRITES_DELTA	NUMBER		Delta value of direct writes
PLSEXEC_TIME_TOTAL	NUMBER		Cumulative value of PL/SQL Execution Time (in microseconds)
PLSEXEC_TIME_DELTA	NUMBER		Delta value of PL/SQL Execution Time (in microseconds)
JAVEXEC_TIME_TOTAL	NUMBER		Cumulative value of Java Execution Time (in microseconds)
JAVEXEC_TIME_DELTA	NUMBER		Delta value of Java Execution Time (in microseconds)
BIND_DATA	RAW(2000)		Bind data

DBA_HIST_SQLTEXT

DBA_HIST_SQLTEXT displays the text of SQL statements belonging to shared SQL cursors captured in the Workload Repository. This view captures information from V$SQL and is used with the DBA_HIST_SQLSTAT view.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SQL_ID	VARCHAR2(13)	NOT NULL	SQL identifier of the parent cursor in the library cache
SQL_TEXT	CLOB		Full text for the SQL statement exposed as a CLOB column
COMMAND_TYPE	NUMBER		Oracle command type definition

DBA_HIST_STAT_NAME

DBA_HIST_STAT_NAME displays decoded statistic names for the statistics captured in the Workload Repository. This view captures information from V$STATNAME and used with DBA_HIST_SYSSTAT and DBA_HIST_SYS_TIME_MODEL.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
STAT_ID	NUMBER	NOT NULL	Statistic identifier
STAT_NAME	VARCHAR2(64)	NOT NULL	Statistic name

DBA_HIST_STREAMS_APPLY_SUM

DBA_HIST_STREAMS_APPLY_SUM displays information about each apply process and its activities. This view contains a snapshot of V$STREAMS_APPLY_COORDINATOR, V$STREAMS_APPLY_READER, and V$STREAMS_APPLY_SERVER. This view is intended for use with Automatic Workload Repository (AWR).

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
APPLY_NAME	VARCHAR2(30)	NOT NULL	Name of the apply process
STARTUP_TIME	DATE	NOT NULL	Time that the apply process was last started
READER_TOTAL_MESSAGES_DEQUEUED	NUMBER		Total number of messages dequeued since the apply process was last started
READER_LAG	NUMBER		For captured messages, the delay (in seconds) between the creation of the last message and it being received by the apply process. For user enqueued messages, the delay between the message being enqueued in the local database and being received by the apply process.
COORD_TOTAL_RECEIVED	NUMBER		Total number of transactions received by the coordinator process since the apply process was last started
COORD_TOTAL_APPLIED	NUMBER		Total number of transactions applied by the apply process since the apply process was last started
COORD_TOTAL_ROLLBACKS	NUMBER		Number of transactions which were rolled back due to unexpected contention
COORD_TOTAL_WAIT_DEPS	NUMBER		Number of times since the apply process was last started that an apply server waited to apply a logical change record (LCR) in a transaction until another apply server applied a transaction because of a dependency between the transactions
COORD_TOTAL_WAIT_CMTS	NUMBER		Number of times since the apply process was last started that an apply server waited to commit a transaction until another apply server committed a transaction to serialize commits
COORD_LWM_LAG	NUMBER		For captured messages, the delay (in seconds) between the creation of the message corresponding to the low-watermark and it being applied by the apply process. For user enqueued messages, the delay between the message being enqueued in the local database and being applied by the apply process.
SERVER_TOTAL_MESSAGES_APPLIED	NUMBER		Total number of messages applied by all the apply servers since the apply process was last started
SERVER_ELAPSED_DEQUEUE_TIME	NUMBER		Time elapsed (in hundredths of a second) dequeuing messages by all the apply servers since the apply process was last started
SERVER_ELAPSED_APPLY_TIME	NUMBER		Time elapsed (in hundredths of a second) applying messages by all the apply servers since the apply process was last started

DBA_HIST_STREAMS_CAPTURE

DBA_HIST_STREAMS_CAPTURE displays information about each capture process. This view contains a snapshot of V$STREAMS_CAPTURE. This view is intended for use with Automatic Workload Repository (AWR).

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
CAPTURE_NAME	VARCHAR2(30)	NOT NULL	Name of the capture process
STARTUP_TIME	DATE	NOT NULL	Time that the capture process was last started
LAG	NUMBER		The delay (in seconds) between the creation and capture of the most recently captured message
TOTAL_MESSAGES_CAPTURED	NUMBER		Total changes captured since the capture process was last started
TOTAL_MESSAGES_ENQUEUED	NUMBER		Total number of messages enqueued since the capture process was last started
ELAPSED_RULE_TIME	NUMBER		Elapsed time (in hundredths of a second) evaluating rules since the capture process was last started
ELAPSED_ENQUEUE_TIME	NUMBER		Elapsed time (in hundredths of a second) enqueuing messages since the capture process was last started
ELAPSED_REDO_WAIT_TIME	NUMBER		Elapsed time (in hundredths of a second) spent by the capture process in the WAITING FOR REDO state
ELAPSED_PAUSE_TIME	NUMBER		Elapsed pause time

DBA_HIST_STREAMS_POOL_ADVICE

DBA_HIST_STREAMS_POOL_ADVICE displays historical information about the estimated count of spilled or unspilled messages and the associated time spent in the spill or unspill activity for different Streams pool sizes. This view is intended for use with Automatic Workload Repository (AWR).

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID of the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number of the snapshot
SIZE_FOR_ESTIMATE	NUMBER	NOT NULL	Pool size for the estimate (in megabytes)
SIZE_FACTOR	NUMBER		Size factor with respect to the current pool size
ESTD_SPILL_COUNT	NUMBER		Estimated count of messages spilled from the Streams pool
ESTD_SPILL_TIME	NUMBER		Estimated elapsed time (in seconds) to spill
ESTD_UNSPILL_COUNT	NUMBER		Estimated count of unspills (read back from disk)
ESTD_UNSPILL_TIME	NUMBER		Estimated elapsed time (in seconds) to unspill

DBA_HIST_SYS_TIME_MODEL

DBA_HIST_SYS_TIME_MODEL displays historical system time model statistics. This view contains snapshots of V$SYS_TIME_MODEL.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic ID
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_SYSMETRIC_HISTORY

DBA_HIST_SYSMETRIC_HISTORY externalizes all available history of the system metric values for the entire set of data kept in the database. This view contains snapshots of V$SYSMETRIC_HISTORY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
VALUE	NUMBER	NOT NULL	Metric Value
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement

DBA_HIST_SYSMETRIC_SUMMARY

DBA_HIST_SYSMETRIC_SUMMARY displays a history of statistical summary of all metric values in the System Metrics Long Duration group. This view contains snapshots of V$SYSMETRIC_SUMMARY.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Group ID
METRIC_ID	NUMBER	NOT NULL	Metric ID
METRIC_NAME	VARCHAR2(64)	NOT NULL	Metric name
METRIC_UNIT	VARCHAR2(64)	NOT NULL	Unit of measurement
NUM_INTERVAL	NUMBER	NOT NULL	Number of intervals observed
MINVAL	NUMBER	NOT NULL	Minimum value observed

Column	Datatype	NULL	Description
MAXVAL	NUMBER	NOT NULL	Maximum value observed
AVERAGE	NUMBER	NOT NULL	Average over the period
STANDARD_DEVIATION	NUMBER	NOT NULL	One standard deviation
SUM_SQUARES	NUMBER		???

DBA_HIST_SYSSTAT

DBA_HIST_SYSSTAT displays historical system statistics information. This view contains snapshots of V$SYSSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
STAT_ID	NUMBER		Statistic identifier
STAT_NAME	VARCHAR2(64)		Statistic name
VALUE	NUMBER		Statistic value

DBA_HIST_SYSTEM_EVENT

DBA_HIST_SYSTEM_EVENT displays historical information on total waits for an event. This view contains snapshots of V$SYSTEM_EVENT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
EVENT_ID	NUMBER	NOT NULL	Identifier of the wait event
EVENT_NAME	VARCHAR2(64)	NOT NULL	Name of the wait event
WAIT_CLASS_ID	NUMBER		Identifier of the Class of the Wait Event
WAIT_CLASS	VARCHAR2(64)		Name of the Class of the Wait Event
TOTAL_WAITS	NUMBER		Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER		Total number of timeouts for the event
TIME_WAITED_MICRO	NUMBER		Total amount of time waited for the event (in microseconds)
TOTAL_WAITS_FG	NUMBER		Total number of waits for the event, from foreground sessions
TOTAL_TIMEOUTS_FG	NUMBER		Total number of timeouts for the event, from foreground sessions
TIME_WAITED_MICRO_FG	NUMBER		Amount of time waited for the event (in microseconds), from foreground sessions

DBA_HIST_TABLESPACE_STAT

DBA_HIST_TABLESPACE_STAT displays tablespace information from the control file. This view contains snapshots of V$TABLESPACE and DBA_TABLESPACE.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
TS#	NUMBER		Tablespace number
TSNAME	VARCHAR2(30)		Tablespace name
CONTENTS	VARCHAR2(9)		Tablespace contents: ■ PERMANENT ■ TEMPORARY
STATUS	VARCHAR2(9)		Tablespace status: ■ ONLINE ■ OFFLINE ■ READ ONLY
SEGMENT_SPACE_ MANAGEMENT	VARCHAR2(6)		Indicates whether the free and used segment space in the tablespace is managed using free lists (MANUAL) or bitmaps (AUTO)
EXTENT_MANAGEMENT	VARCHAR2(10)		Indicates whether the extents in the tablespace are dictionary managed (DICTIONARY) or locally managed (LOCAL)
IS_BACKUP	VARCHAR2(5)		Indicates whether the tablespace is part of a backup

DBA_HIST_TBSPC_SPACE_USAGE

DBA_HIST_TBSPC_SPACE_USAGE displays historical tablespace usage statistics.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
TABLESPACE_ID	NUMBER		Tablespace ID
TABLESPACE_SIZE	NUMBER		Tablespace size
TABLESPACE_MAXSIZE	NUMBER		Maximum size of the tablespace
TABLESPACE_USEDSIZE	NUMBER		Used size of the tablespace
RTIME	VARCHAR2(25)		Runtime

DBA_HIST_TEMPFILE

DBA_HIST_TEMPFILE displays a history of the tempfile information from the control file. This view contains snapshots of V$TEMPFILE.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the tempfile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the tempfile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the tempfile

DBA_HIST_TEMPSTATXS

DBA_HIST_TEMPSTATXS displays information about temporary file read/write statistics. This view contains snapshots of V$TEMPSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
FILE#	NUMBER	NOT NULL	File identification number
CREATION_CHANGE#	NUMBER	NOT NULL	Change number at which the tempfile was created
FILENAME	VARCHAR2(513)	NOT NULL	Name of the tempfile
TS#	NUMBER	NOT NULL	Tablespace number
TSNAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER		Block size of the tempfile
PHYRDS	NUMBER		Number of physical reads done
PHYWRTS	NUMBER		Number of times DBWR is required to write
SINGLEBLKRDS	NUMBER		Number of single block reads
READTIM	NUMBER		Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER		Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER		Cumulative single block read time (in hundredths of a second)
PHYBLKRD	NUMBER		Number of physical blocks read
PHYBLKWRT	NUMBER		Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
WAIT_COUNT	NUMBER		Wait Count
TIME	NUMBER		Wait time

DBA_HIST_THREAD

DBA_HIST_THREAD displays historical thread information from the control file.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
THREAD#	NUMBER	NOT NULL	Thread number
THREAD_INSTANCE_NUMBER	NUMBER		Instance number of the thread
STATUS	VARCHAR2(6)		Thread status (OPEN) or (CLOSED)
OPEN_TIME	DATE		Last time the thread was opened
CURRENT_GROUP#	NUMBER		Current log group
SEQUENCE#	NUMBER		Sequence number of the current log

DBA_HIST_UNDOSTAT

DBA_HIST_UNDOSTAT displays the history of histograms of statistical data to show how well the system is working. The available statistics include undo space consumption, transaction concurrency, and length of queries executed in the instance. This view contains snapshots of V$UNDOSTAT.

Column	Datatype	NULL	Description
BEGIN_TIME	DATE	NOT NULL	Identifies the beginning of the time interval
END_TIME	DATE	NOT NULL	Identifies the end of the time interval
DBID	NUMBER	NOT NULL	Database ID for the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number for the snapshot
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
UNDOTSN	NUMBER	NOT NULL	Represents the last active undo tablespace in the duration of time. The tablespace ID of the active undo tablespace is returned in this column. If more than one undo tablespace was active in that period, the active undo tablespace that was active at the end of the period is reported.
UNDOBLKS	NUMBER		Represents the total number of undo blocks consumed. You can use this column to obtain the consumption rate of undo blocks, and thereby estimate the size of the undo tablespace needed to handle the workload on your system.
TXNCOUNT	NUMBER		Identifies the total number of transactions executed within the period
MAXQUERYLEN	NUMBER		Identifies the length of the longest query (in number of seconds) executed in the instance during the period. You can use this statistic to estimate the proper setting of the UNDO_RETENTION initialization parameter. The length of a query is measured from the cursor open time to the last fetch/execute time of the cursor. Only the length of those cursors that have been fetched/executed during the period are reflected in the view.
MAXQUERYSQLID	VARCHAR2(13)		SQL identifier of the longest running SQL statement in the period
MAXCONCURRENCY	NUMBER		Identifies the highest number of transactions executed concurrently within the period
UNXPSTEALCNT	NUMBER		Number of attempts to obtain undo space by stealing unexpired extents from other transactions
UNXPBLKRELCNT	NUMBER		Number of unexpired blocks removed from certain undo segments so they can be used by other transactions
UNXPBLKREUCNT	NUMBER		Number of unexpired undo blocks reused by transactions
EXPSTEALCNT	NUMBER		Number of attempts to steal expired undo blocks from other undo segments
EXPBLKRELCNT	NUMBER		Number of expired undo blocks stolen from other undo segments
EXPBLKREUCNT	NUMBER		Number of expired undo blocks reused within the same undo segments
SSOLDERRCNT	NUMBER		Identifies the number of times the error ORA-01555 occurred. You can use this statistic to decide whether or not the UNDO_RETENTION initialization parameter is set properly given the size of the undo tablespace. Increasing the value of UNDO_RETENTION can reduce the occurrence of this error.

Column	Datatype	NULL	Description
NOSPACEERRCNT	NUMBER		Identifies the number of times space was requested in the undo tablespace and there was no free space available. That is, all of the space in the undo tablespace was in use by active transactions. The corrective action is to add more space to the undo tablespace.
ACTIVEBLKS	NUMBER		Total number of blocks in the active extents of the undo tablespace for the instance at the sampled time in the period
UNEXPIREDBLKS	NUMBER		Total number of blocks in the unexpired extents of the undo tablespace for the instance at the sampled time in the period
EXPIREDBLKS	NUMBER		Total number of blocks in the expired extents of the undo tablespace for the instance at the sampled time in the period
TUNED_UNDORETENTION	NUMBER		System tuned value indicating the period for which undo is being retained

DBA_HIST_WAITCLASSMET_HISTORY

DBA_HIST_WAITCLASSMET_HISTORY displays the history of the wait event class metric data kept by the Workload Repository.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER	NOT NULL	Unique snapshot ID
DBID	NUMBER	NOT NULL	Database ID of the snapshot
INSTANCE_NUMBER	NUMBER	NOT NULL	Instance number of the snapshot
WAIT_CLASS_ID	NUMBER	NOT NULL	Identifier of the class of the wait event
WAIT_CLASS	VARCHAR2(64)		Name of the class of the wait event
BEGIN_TIME	DATE	NOT NULL	Begin time of the interval
END_TIME	DATE	NOT NULL	End time of the interval
INTSIZE	NUMBER	NOT NULL	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	NOT NULL	Metric group ID
AVERAGE_WAITER_COUNT	NUMBER	NOT NULL	Average waiter count
DBTIME_IN_WAIT	NUMBER	NOT NULL	Percent of database time spent in the wait
TIME_WAITED	NUMBER	NOT NULL	Time waited during the interval (in microseconds)
WAIT_COUNT	NUMBER	NOT NULL	Number of times waited

DBA_HIST_WAITSTAT

DBA_HIST_WAITSTAT displays historical block contention statistics. This view contains snapshots of V$WAITSTAT.

Column	Datatype	NULL	Description
SNAP_ID	NUMBER		Unique snapshot ID
DBID	NUMBER		Database ID for the snapshot
INSTANCE_NUMBER	NUMBER		Instance number for the snapshot
CLASS	VARCHAR2(18)		Class of the block
WAIT_COUNT	NUMBER		Number of waits by the OPERATION for this CLASS of block

Column	Datatype	NULL	Description
TIME	NUMBER		Sum of all wait times for all the waits by the OPERATION for this CLASS of block

DBA_HIST_WR_CONTROL

DBA_HIST_WR_CONTROL displays the control information for the Workload Repository.

Column	Datatype	NULL	Description
DBID	NUMBER	NOT NULL	Database ID
SNAP_INTERVAL	INTERVAL DAY(5) TO SECOND(1)	NOT NULL	Snapshot interval; how often to automatically take snapshots
RETENTION	INTERVAL DAY(5) TO SECOND(1)	NOT NULL	Retention setting for the snapshots; amount of time to keep the snapshots
TOPNSQL	VARCHAR2(10)		The number of Top SQL flushed for each SQL criteria (elapsed time, CPU time, parse calls, shareable memory, version count)

DBA_IDENTIFIERS

DBA_IDENTIFIERS displays information about the identifiers in all stored objects in the database. Its columns are the same as those in ALL_IDENTIFIERS.

See Also: "ALL_IDENTIFIERS" on page 2-57

DBA_IND_COLUMNS

DBA_IND_COLUMNS describes the columns of all the indexes on all tables and clusters in the database. Its columns are the same as those in "ALL_IND_COLUMNS" on page 2-58.

DBA_IND_EXPRESSIONS

DBA_IND_EXPRESSIONS lists expressions of function-based indexes on all tables and clusters in the database. Its columns are the same as those in "ALL_IND_EXPRESSIONS" on page 2-59.

DBA_IND_PARTITIONS

DBA_IND_PARTITIONS describes, for each index partition in the database, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_PARTITIONS" on page 2-59.

DBA_IND_PENDING_STATS

DBA_IND_PENDING_STATS describes pending statistics for all tables, partitions, and subpartitions in the database. Its columns are the same as those in "ALL_IND_PENDING_STATS" on page 2-60.

DBA_IND_STATISTICS

DBA_IND_STATISTICS displays optimizer statistics for all indexes in the database. Its columns are the same as those in ALL_IND_STATISTICS.

See Also: "ALL_IND_STATISTICS" on page 2-61

DBA_IND_SUBPARTITIONS

DBA_IND_SUBPARTITIONS describes, for each index subpartition in the database, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_SUBPARTITIONS" on page 2-62.

DBA_INDEXES

DBA_INDEXES describes all indexes in the database. To gather statistics for this view, use the SQL ANALYZE statement. This view supports parallel partitioned index scans. Its columns are the same as those in "ALL_INDEXES" on page 2-64.

DBA_INDEXTYPE_ARRAYTYPES

DBA_INDEXTYPE_ARRAYTYPES displays information about the array types specified by all indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_ARRAYTYPES.

See Also: "ALL_INDEXTYPE_ARRAYTYPES" on page 2-66

DBA_INDEXTYPE_COMMENTS

DBA_INDEXTYPE_COMMENTS displays comments for all user-defined indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_COMMENTS.

See Also: "ALL_INDEXTYPE_COMMENTS" on page 2-67

DBA_INDEXTYPE_OPERATORS

DBA_INDEXTYPE_OPERATORS lists all the operators supported by indextypes in the database. Its columns are the same as those in ALL_INDEXTYPE_OPERATORS.

See Also: "ALL_INDEXTYPE_OPERATORS" on page 2-67

DBA_INDEXTYPES

DBA_INDEXTYPES describes all indextypes in the database. Its columns are the same as those in ALL_INDEXTYPES.

See Also: "ALL_INDEXTYPES" on page 2-68

DBA_INTERNAL_TRIGGERS

DBA_INTERNAL_TRIGGERS describes internal triggers on all tables in the database. Its columns are the same as those in "ALL_INTERNAL_TRIGGERS" on page 2-69.

DBA_JAVA_ARGUMENTS

DBA_JAVA_ARGUMENTS displays argument information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_ARGUMENTS.

See Also: "ALL_JAVA_ARGUMENTS" on page 2-69

DBA_JAVA_CLASSES

DBA_JAVA_CLASSES displays class level information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_CLASSES.

See Also: "ALL_JAVA_CLASSES" on page 2-70

DBA_JAVA_COMPILER_OPTIONS

DBA_JAVA_COMPILER_OPTIONS displays information about all native compiler options in the database. Its columns are the same as those in ALL_JAVA_COMPILER_OPTIONS.

See Also: "ALL_JAVA_COMPILER_OPTIONS" on page 2-70

DBA_JAVA_DERIVATIONS

DBA_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for all Java classes in the database. Its columns are the same as those in ALL_JAVA_DERIVATIONS.

See Also: "ALL_JAVA_DERIVATIONS" on page 2-71

DBA_JAVA_FIELDS

DBA_JAVA_FIELDS displays field information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_FIELDS.

See Also: "ALL_JAVA_FIELDS" on page 2-71

DBA_JAVA_IMPLEMENTS

DBA_JAVA_IMPLEMENTS describes interfaces implemented by all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_IMPLEMENTS.

See Also: "ALL_JAVA_IMPLEMENTS" on page 2-72

DBA_JAVA_INNERS

DBA_JAVA_INNERS displays information about inner classes referred to by all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_INNERS.

See Also: "ALL_JAVA_INNERS" on page 2-73

DBA_JAVA_LAYOUTS

DBA_JAVA_LAYOUTS displays class layout information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_LAYOUTS.

See Also: "ALL_JAVA_LAYOUTS" on page 2-74

DBA_JAVA_METHODS

DBA_JAVA_METHODS displays method information about all stored Java classes in the database. Its columns are the same as those in ALL_JAVA_METHODS.

See Also: "ALL_JAVA_METHODS" on page 2-74

DBA_JAVA_NCOMPS

DBA_JAVA_NCOMPS displays ncomp-related information about all Java classes in the database. Its columns are the same as those in ALL_JAVA_NCOMPS.

See Also: "ALL_JAVA_NCOMPS" on page 2-75

DBA_JAVA_POLICY

DBA_JAVA_POLICY describes Java security permissions for all users in the database.

Related View

USER_JAVA_POLICY describes Java security permissions for the current user.

Column	Datatype	NULL	Description
KIND	VARCHAR2(8)		Indicates whether the permission is a positive (GRANT) or a limitation (RESTRICT)
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user, schema, or role to which the permission object is assigned
TYPE_SCHEMA	VARCHAR2(30)	NOT NULL	Schema in which the permission object is loaded
TYPE_NAME	VARCHAR2(4000)		Permission class type, which is designated by a string containing the full class name, such as, java.io.FilePermission
NAME	VARCHAR2(4000)		Target attribute (name) of the permission object. This name is used when defining the permission.
ACTION	VARCHAR2(4000)		Action attribute for this permission. Many permissions expect a null value if no action is appropriate for the permission.
ENABLED	VARCHAR2(8)		Indicates whether the permission is enabled (ENABLED) or disabled (DISABLED)
SEQ	NUMBER		Sequence number used to identify this row. This number should be supplied when disabling, enabling, or deleting the permission.

See Also: "USER_JAVA_POLICY" on page 5-114

DBA_JAVA_RESOLVERS

DBA_JAVA_RESOLVERS displays information about resolvers of all Java classes in the database. Its columns are the same as those in ALL_JAVA_RESOLVERS.

See Also: "ALL_JAVA_RESOLVERS" on page 2-76

DBA_JAVA_THROWS

DBA_JAVA_THROWS displays information about exceptions thrown from methods of all Java classes in the database. Its columns are the same as those in ALL_JAVA_THROWSa.

See Also: "ALL_JAVA_THROWS" on page 2-76

DBA_JOBS

DBA_JOBS describes all jobs in the database.

Related View

USER_JOBS describes the jobs owned by the current user.

Column	Datatype	NULL	Description
JOB	NUMBER	NOT NULL	Identifier of job. Neither import/export nor repeated executions change this value.
LOG_USER	VARCHAR2(30)	NOT NULL	Login user when the job was submitted
PRIV_USER	VARCHAR2(30)	NOT NULL	User whose default privileges apply to this job
SCHEMA_USER	VARCHAR2(30)	NOT NULL	Default schema used to parse the job For example, if the SCHEMA_USER is SCOTT and you submit the procedure HIRE_EMP as a job, the Oracle Database looks for SCOTT.HIRE_EMP
LAST_DATE	DATE		Date on which this job last successfully executed
LAST_SEC	VARCHAR2(8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing (usually null if not executing)
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE. This is when the last successful execution started.
NEXT_DATE	DATE	NOT NULL	Date that this job will next be executed
NEXT_SEC	VARCHAR2(8)		Same as NEXT_DATE. This is when the last successful execution started.
TOTAL_TIME	NUMBER		Total wall clock time spent by the system on this job (in seconds) since the first time this job executed. This value is cumulative.
BROKEN	VARCHAR2(1)		Y: no attempt is made to run this job N: an attempt is made to run this job
INTERVAL	VARCHAR2(200)	NOT NULL	A date function, evaluated at the start of execution, becomes next NEXT_DATE
FAILURES	NUMBER		Number of times the job has started and failed since its last success
WHAT	VARCHAR2(4000)		Body of the anonymous PL/SQL block that the job executes
NLS_ENV	VARCHAR2(4000)		Session parameters describing the NLS environment of the job
MISC_ENV	RAW(32)		Other session parameters that apply to this job
INSTANCE	NUMBER		ID of the instance that can execute or is executing the job. The default is 0.

See Also: "USER_JOBS" on page 5-114

DBA_JOBS_RUNNING

DBA_JOBS_RUNNING lists all jobs that are currently running in the instance.

Column	Datatype	NULL	Description
SID	NUMBER		Identifier of process that is executing the job. See "V$LOCK" on page 7-13.
JOB	NUMBER		Identifier of job. This job is currently executing.
FAILURES	NUMBER		Number of times this job started and failed since its last success.
LAST_DATE	DATE		Date that this job last successfully executed.
LAST_SEC	VARCHAR2(8)		Same as LAST_DATE. This is when the last successful execution started.
THIS_DATE	DATE		Date that this job started executing.
THIS_SEC	VARCHAR2(8)		Same as THIS_DATE. This is when the last successful execution started.
INSTANCE	NUMBER		Indicates which instance can execute or is executing the job; the default is 0.

DBA_JOIN_IND_COLUMNS

DBA_JOIN_IND_COLUMNS describes all join conditions in the database. Its columns are the same as those in "ALL_JOIN_IND_COLUMNS" on page 2-77.

DBA_KGLLOCK

DBA_KGLLOCK lists all the locks and pins held on KGL objects (objects in the Kernel Generic Library cache).

Column	Datatype	NULL	Description
kgllkuse	RAW(4)		Address of the user session that holds the lock or pin
kgllkhdl	RAW(4)		Address of the handle for the KGL object
kgllkmod	NUMBER		Current mode of the lock or pin
kgllkreq	NUMBER		Mode in which the lock or pin was requested
kgllktype	VARCHAR2(4)		Whether this is a lock or a pin

DBA_LIBRARIES

DBA_LIBRARIES describes all libraries in the database. Its columns are the same as those in ALL_LIBRARIES.

See Also: "ALL_LIBRARIES" on page 2-77

DBA_LMT_FREE_SPACE

DBA_LMT_FREE_SPACE describes the free extents in all locally managed tablespaces in the database.

Column	Datatype	NULL	Description
TABLESPACE_ID	NUMBER		Identifier number of the tablespace containing the extent
FILE_ID	NUMBER		File identifier number of the file containing the extent

Column	Datatype	NULL	Description
BLOCK_ID	NUMBER		Starting block number of the extent
BLOCKS	NUMBER		Size of the extent (in Oracle blocks)

DBA_LMT_USED_EXTENTS

DBA_LMT_USED_EXTENTS describes the extents comprising the segments in all locally managed tablespaces in the database.

Column	Datatype	NULL	Description
SEGMENT_FILEID	NUMBER		File number of the segment header of the extent
SEGMENT_BLOCK	NUMBER		Block number of the segment header of the extent
TABLESPACE_ID	NUMBER		Identifier number of the tablespace containing the extent
EXTENT_ID	NUMBER		Extent number in the segment
FILEID	NUMBER		File identifier number of the file containing the extent
BLOCK	NUMBER		Starting block number of the extent
LENGTH	NUMBER		Number of blocks in the extent

DBA_LOB_PARTITIONS

DBA_LOB_PARTITIONS displays all LOB partitions in the database. Its columns are the same as those in "ALL_LOB_PARTITIONS" on page 2-78.

DBA_LOB_SUBPARTITIONS

DBA_LOB_SUBPARTITIONS displays partition-level attributes of all LOB data subpartitions in the database. Its columns are the same as those in "ALL_LOB_SUBPARTITIONS" on page 2-79.

DBA_LOB_TEMPLATES

DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database. Its columns are the same as those in ALL_LOB_TEMPLATES.

See Also: "ALL_LOB_TEMPLATES" on page 2-80

DBA_LOBS

DBA_LOBS displays the BLOBs and CLOBs contained in all tables in the database. BFILEs are stored outside the database, so they are not described by this view. This view's columns are the same as those in "ALL_LOBS" on page 2-81.

DBA_LOCK

DBA_LOCK lists all locks or latches held in the database, and all outstanding requests for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock

Column	Datatype	NULL	Description
LOCK_TYPE	VARCHAR2(26)		Lock type **See Also:** For a listing of lock types, see Appendix D, "Oracle Enqueue Names"
MODE HELD	VARCHAR2(40)		Lock mode
MODE REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(40)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2
LAST_CONVERT	NUMBER		The last convert
BLOCKING_OTHERS	VARCHAR2(40)		Whether the lock is currently blocking others

DBA_LOCK_INTERNAL

DBA_LOCK_INTERNAL displays a row for each lock or latch that is being held, and one row for each outstanding request for a lock or latch.

Column	Datatype	NULL	Description
SESSION_ID	NUMBER		Session holding or acquiring the lock
LOCK_TYPE	VARCHAR2(56)		Lock type **See Also:** For a listing of lock types, see Appendix D, "Oracle Enqueue Names"
MODE HELD	VARCHAR2(40)		Lock mode
MODE REQUESTED	VARCHAR2(40)		Lock mode requested
LOCK_ID1	VARCHAR2(1130)		Type-specific lock identifier, part 1
LOCK_ID2	VARCHAR2(40)		Type-specific lock identifier, part 2

DBA_LOCKS

DBA_LOCKS is a synonym for DBA_LOCK.

See Also: "DBA_LOCK" on page 4-141

DBA_LOG_GROUP_COLUMNS

DBA_LOG_GROUP_COLUMNS describes all columns in the database that are specified in log groups. Its columns are the same as those in "ALL_LOG_GROUP_COLUMNS" on page 2-81.

DBA_LOG_GROUPS

DBA_LOG_GROUPS describes log group definitions on all tables in the database. Its columns are the same as those in "ALL_LOG_GROUPS" on page 2-82.

DBA_LOGMNR_LOG

DBA_LOGMNR_LOG displays all archived logs registered with active LogMiner persistent sessions in the database.

A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.

Column	Datatype	NULL	Description
LOGMNR_SESSION_ID	NUMBER	NOT NULL	Unique identifier of the persistent session
NAME	VARCHAR2(513)		Name of the archived log
DBID	NUMBER	NOT NULL	Database identifier that produced the archived log
RESETLOGS_SCN	NUMBER	NOT NULL	SCN at which resetlogs operation was performed at the source database generating the archived log
RESETLOGS_TIME	NUMBER	NOT NULL	Timestamp at which resetlogs operation was performed at the source database generating the archived log
MODIFIED_TIME	DATE		Time at which the archived log was registered with LogMiner
THREAD#	NUMBER	NOT NULL	Redo thread at the source database that generated the archived log
SEQUENCE#	NUMBER	NOT NULL	Logfile sequence number
FIRST_SCN	NUMBER	NOT NULL	Lowest SCN of the redo record contained in the logfile
NEXT_SCN	NUMBER	NOT NULL	Highest possible SCN of the redo record contained in the logfile
FIRST_TIME	DATE		Time of the first redo record contained in the logfile
NEXT_TIME	DATE		Time of the last redo record contained in the logfile
DICTIONARY_BEGIN	VARCHAR2(3)		Indicates whether the archived log contains the beginning of a LogMiner dictionary (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)		Indicates whether the archived log contains the end of a LogMiner dictionary (YES) or not (NO)
KEEP	VARCHAR2(3)		Indicates whether the logfile is still required for this LogMiner session (YES) or not (NO)
SUSPECT	VARCHAR2(3)		Indicates whether the archived log content was deemed to be corrupt or the archived log is partially filled (YES) or not (NO)

DBA_LOGMNR_PURGED_LOG

DBA_LOGMNR_PURGED_LOG displays archived redo log files that have been applied to the logical standby database and can be deleted because they are no longer needed. FIles in this view are refreshed as a result of executing either of the following PL/SQL procedures:

- DBMS_LOGSTDBY.PURGE_SESSION (for Oracle Data Guard SQL Apply)
- DBMS_CAPTURE.SET_FIRST_SCN (for Oracle Streams)

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Fully qualified names of the archived redo log files that are no longer needed by SQL Apply and can be deleted from the operating system

DBA_LOGMNR_SESSION

DBA_LOGMNR_SESSION displays all active LogMiner persistent sessions in the database.

A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Unique session identifier

Column	Datatype	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	Unique session name
SOURCE_DATABASE	VARCHAR2(128)		Global name of the source database whose archived logs are to be mined in this persistent LogMiner session
SOURCE_DBID	NUMBER		Database ID of the source database
SOURCE_RESETLOGS_SCN	NUMBER		Resetlogs SCN associated with the incarnation of the source database whose archived logs are mined
SOURCE_RESETLOGS_TIME	NUMBER		Resetlogs time associated with the incarnation of the source database whose archived logs are mined
FIRST_SCN	NUMBER		Only modifications that occurred on or after this SCN can be mined using this persistent session
END_SCN	NUMBER		No modifications that occurred on or after this SCN can be mined using this persistent session
BRANCH_SCN	NUMBER		SCN at which a branch will be taken in terms of the incarnation corresponding to the source database. This implies a point-in-time recovery was performed at the source database at this SCN.
WAIT_FOR_LOG	VARCHAR2(3)		Indicates whether the persistent session waits for RFS to register new archived logs or to fill gaps (YES) or not (NO)
HOT_MINE	VARCHAR2(3)		Indicates whether real-time mining is on (YES) or not (NO)
SAFE_PURGE_SCN	NUMBER		Persistent session can safely be purged up to this SCN
CHECKPOINT_SCN	NUMBER		SCN at which the latest checkpoint is taken by the persistent LogMiner session

DBA_LOGSTDBY_EVENTS

DBA_LOGSTDBY_EVENTS displays information about the activity of the logical standby database system. It can be used to determine the cause of failures that occur when applying redo data to logical standby databases. This view is for logical standby databases only.

Column	Datatype	NULL	Description
EVENT_TIME	DATE		Time when the event was logged
EVENT_TIMESTAMP	TIMESTAMP(6)		Timestamp when the event was logged
CURRENT_SCN	NUMBER		SCN associated with the change at the primary database. If a failure occurred, then examine this column to determine which archived log file contains the source of the failure (for example, an unsupported record).
COMMIT_SCN	NUMBER		SCN value on which the change was committed at the primary database
XIDUSN	NUMBER		Transaction ID undo segment number at the primary database of the associated transaction
XIDSLT	NUMBER		Transaction ID slot number at the primary database of the associated transaction
XIDSQN	NUMBER		Transaction ID sequence number at the primary database of the associated transaction
EVENT	CLOB		Statement that was being processed when the failure occurred
STATUS_CODE	NUMBER		Status (or Oracle error code) belonging to the STATUS message
STATUS	VARCHAR2(2000)		Description of the current activity of the process or the reason why the apply operation stopped

DBA_LOGSTDBY_HISTORY

DBA_LOGSTDBY_HISTORY displays the history of switchovers and failovers in a Data Guard configuration. It does this by showing the complete sequence of redo log streams processed or created on the local system, across all role transitions. (After a role transition, a new log stream is started and the log stream sequence number is incremented by the new primary database.) This view is for logical standby databases only.

Column	Datatype	NULL	Description
STREAM_SEQUENCE#	NUMBER		Lists the sequence numbers for all log streams created or applied on the local system. **Note**: A value of 0 indicates an unknown sequence order; this is reserved for future log streams.
STATUS	VARCHAR2(16)		Description of the log stream processing: ■ Past - The log stream has already been processed ■ Immediate Past - This is the most recently processed log stream; its status is transitioning from Current to Past ■ Current - The log stream is currently being processed ■ Immediate Future - This is the next log stream to be processed; its status is transitioning from Future to Current ■ Future - The log stream will be processed
SOURCE	VARCHAR2(5)		Describes how the log stream was started: ■ RFS - The RFS process created the log stream ■ User - A user registered the initial log file for the log stream ■ Synch - A user issued the ALTER DATABASE START LOGICAL STANDBY APPLY NEW PRIMARY DDL statement ■ Redo - The log stream information was recorded in the redo log
DBID	NUMBER		Database identifier of the primary database that created the log stream
FIRST_CHANGE#	NUMBER		Lowest system change number (SCN) in the current log file
LAST_CHANGE#	NUMBER		Highest system change number (SCN) in the current log file
FIRST_TIME	DATE		Time of the first SCN entry (FIRST_CHANGE#) in the current log file
LAST_TIME	DATE		Time of the last SCN entry (LAST_CHANGE#) in the current log file
DGNAME	VARCHAR2(255)		Unique database name (DB_UNIQUE_NAME) of the primary database that produced the log stream. See V$DATAGUARD_CONFIG to display all database DB_UNIQUE_NAME values defined in the Data Guard configuration.
MERGE_CHANGE#	NUMBER		SCN that should be used to flashback a failed primary (that created the log stream) or to flashback a bystander logical standby database following a failover, in the context of the associated redo log stream. It is the SCN up to which redo for the associated log stream can be merged safely in all databases using local copies of archived logs received from the primary database. In order to apply changes beyond this following a failover, you will need to fetch and mine the redo logs from the failover target.

Column	Datatype	NULL	Description
PROCESSED_CHANGE#	NUMBER		Strict upper bound on the SCN up to which SQL Apply has applied redo records before it switched to a new log stream (either because it was activated and became the primary database, or in the case of a bystander logical standby database where it switched to a new log stream in order to accommodate a new primary database).

DBA_LOGSTDBY_LOG

DBA_LOGSTDBY_LOG displays information about the logs registered for a logical standby database. This view is for logical standby databases only.

Column	Datatype	NULL	Description
THREAD#	NUMBER	NOT NULL	Thread ID of the archive log. The THREAD number is 1 for a single instance. For Real Application Clusters, this column will contain different numbers.
RESETLOGS_CHANGE#	NUMBER	NOT NULL	The start SCN of the branch
RESETLOGS_ID	NUMBER	NOT NULL	Resetlogs identifier (a numerical form of the timestamp of the branch)
SEQUENCE#	NUMBER	NOT NULL	Sequence number of the archive log file
FIRST_CHANGE#	NUMBER	NOT NULL	System change number (SCN) of the current archive log
NEXT_CHANGE#	NUMBER	NOT NULL	SCN of the next archive log
FIRST_TIME	DATE		Date of the current archive log
NEXT_TIME	DATE		Date of the next archive log
FILE_NAME	VARCHAR2(513)		Name of the archive log
TIMESTAMP	DATE		Time when the archive log was registered
DICT_BEGIN	VARCHAR2(3)		Indicates whether the beginning of the dictionary build is in this archive log (YES) or not (NO)
DICT_END	VARCHAR2(3)		Indicates whether the end of the dictionary build is in this archive log (YES) or not (NO)
APPLIED	VARCHAR2(7)		Indicates whether each archive log has been applied (YES) or not (NO)

Note: The SCN values in this view correlate to the SCN values shown in the DBA_LOGSTDBY_PROGRESS view.

DBA_LOGSTDBY_NOT_UNIQUE

DBA_LOGSTDBY_NOT_UNIQUE displays all tables that have no primary and no non-null unique indexes. Most of the tables displayed by this view are supported because their columns contain enough information to be maintained in a logical standby database. Some tables, however, cannot be supported because their columns do not contain the necessary information. Unsupported tables usually contain a column defined using an unsupported datatype.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Schema name of the non-unique table
TABLE_NAME	VARCHAR2(30)		Table name of the non-unique table

Column	Datatype	NULL	Description
BAD_COLUMN	VARCHAR2(1)		■ Y - Table column is defined using an unbounded data type, such as LONG or BLOB. If two rows in the table match except in their LOB columns, then the table cannot be maintained properly. Log apply services will attempt to maintain these tables, but you must ensure the application does not allow uniqueness only in the unbounded columns. ■ N - Enough column information is present to maintain the table in the logical standby database but the log transport services and log apply services would run more efficiently if you added a primary key. You should consider adding a disabled RELY constraint to these tables.

DBA_LOGSTDBY_PARAMETERS

DBA_LOGSTDBY_PARAMETERS displays the list of parameters used by SQL apply for logical standby databases. This view is for logical standby databases only.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the parameter: ■ MAX_SGA - System global area (SGA) allocated for the log apply services cache (in megabytes) ■ MAX_SERVERS - Number of processes used by SQL Apply services ■ PREPARE_SERVERS - Controls the number of parallel execution servers used to prepare changes ■ APPLY_SERVERS - Controls the number of parallel execution servers used to apply changes ■ MAX_EVENTS_RECORDED - Number of events stored in the DBA_LOGSTDBY_EVENTS view ■ RECORD_SKIP_ERRORS - Indicates records that are skipped ■ RECORD_SKIP_DDL - Indicates skipped DDL statements ■ RECORD_APPLIED_DDL - Indicates applied DDL statements ■ RECORD_UNSUPPORTED_OPERATIONS - Shows whether SQL Apply will capture information about transactions that did unsupported operations at the primary database in the DBA_LOGSTDBY_EVENTS view ■ EVENT_LOG_DEST - ??? ■ LOG_AUTO_DELETE - Shows whether SQL Apply will automatically delete remote archived logs received from the primary database, once the contents of the logs are applied at the logical standby database. ■ LOG_AUTO_DEL_RETENTION_TARGET - How many minutes a remote archived log received from the primary database will be retained at the logical standby database, once the contents of the log are applied by SQL Apply. ■ PRESERVE_COMMIT_ORDER - Shows whether transactions are committed at the logical standby database in the same order that they were committed at the primary database
VALUE	VARCHAR2(2000)		Value of the parameter
UNIT	VARCHAR2(64)		Unit of the value, if applicable

Column	Datatype	NULL	Description
SETTING	VARCHAR2(64)		Possible values are as follows: ■ SYSTEM - Parameter value was not explicitly set by the user. However, the user can change it with an appropriate call to the APPLY_SET procedure. ■ USER - Parameter value was explicitly set by the user
DYNAMIC	VARCHAR2(64)		YES if the parameter can be set dynamically (that is, without having to stop SQL Apply) NO if setting the parameter requires that SQL Apply be stopped

DBA_LOGSTDBY_PROGRESS

DBA_LOGSTDBY_PROGRESS is deprecated. The information that was provided in this view is now provided in the V$LOGSTDBY_PROGRESS view.

DBA_LOGSTDBY_SKIP

DBA_LOGSTDBY_SKIP displays the tables that will be skipped by log apply services. This view is for logical standby databases only.

Column	Datatype	NULL	Description
ERROR	VARCHAR2(1)		Indicates whether the statement should be skipped (Y) or just return errors for the statement (N)
STATEMENT_OPT	VARCHAR2(30)		Specifies the type of statement that should be skipped. It must be one of the SYSTEM_AUDIT statement options.
OWNER	VARCHAR2(30)		Name of the schema under which the skip option should be used
NAME	VARCHAR2(65)		Name of the table that is being skipped
USE_LIKE	VARCHAR2(1)		Indicates whether the statement should use a SQL wildcard search when matching names (Y) or not (N)
ESC	VARCHAR2(1)		Escape character used when performing wildcard matches
PROC	VARCHAR2(98)		Name of a stored procedure that will be executed when processing the skip option

DBA_LOGSTDBY_SKIP_TRANSACTION

DBA_LOGSTDBY_SKIP_TRANSACTION displays the skip settings chosen. This view is for logical standby databases only.

Column	Datatype	NULL	Description
XIDUSN	NUMBER		Transaction ID undo segment number
XIDSLT	NUMBER		Transaction ID slot number
XIDSQN	NUMBER		Transaction ID sequence number

DBA_LOGSTDBY_UNSUPPORTED

DBA_LOGSTDBY_UNSUPPORTED displays the schemas, tables, and columns in those tables that contain unsupported datatypes. Use this view when you are preparing to create a logical standby database. This view is for logical standby databases only.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the unsupported table
TABLE_NAME	VARCHAR2(30)		Name of the unsupported table
COLUMN_NAME	VARCHAR2(30)		Name of the unsupported column
ATTRIBUTES	VARCHAR2(39)		Displays the reason why the table is not supported by SQL Apply. If the structure of the table is unsupported (not just the datatype of some columns), then this view displays all columns in the table, with one of the following values displayed for each column in the table: ■ IOT with Overflow ■ IOT with LOB ■ Mapping table for physical rowid of IOT ■ IOT with row movement ■ Table Compression ■ Object Table ■ AQ queue table ■ Unsupported Virtual Column ■ Encrypted Column If the structure of the table is supported but certain columns in the table have unsupported datatypes, the ATTRIBUTE column will be NULL.
DATA_TYPE	VARCHAR2(32)		Datatype of the unsupported column

DBA_LOGSTDBY_UNSUPPORTED_TABLE

DBA_LOGSTDBY_UNSUPPORTED_TABLE displays the data tables that are not supported by Logical Standby. This view is for logical standby databases only.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the unsupported table
TABLE_NAME	VARCHAR2(30)		Name of the unsupported table

DBA_MEASURE_FOLDER_CONTENTS

DBA_MEASURE_FOLDER_CONTENTS describes the contents of all OLAP measure folders in the database. Its columns are the same as those in ALL_MEASURE_FOLDER_CONTENTS.

See Also: "ALL_MEASURE_FOLDER_CONTENTS" on page 2-82

DBA_MEASURE_FOLDERS

DBA_MEASURE_FOLDERS describes all OLAP measure folders in the database. Its columns are the same as those in ALL_MEASURE_FOLDERS.

See Also: "ALL_MEASURE_FOLDERS" on page 2-83

DBA_METHOD_PARAMS

DBA_METHOD_PARAMS describes the method parameters of all object types in the database. Its columns are the same as those in ALL_METHOD_PARAMS.

See Also: "ALL_METHOD_PARAMS" on page 2-83

DBA_METHOD_RESULTS

DBA_METHOD_RESULTS describes the method results of all object types in the database. Its columns are the same as those in ALL_METHOD_RESULTS.

See Also: "ALL_METHOD_RESULTS" on page 2-84

DBA_MINING_MODEL_ATTRIBUTES

DBA_MINING_MODEL_ATTRIBUTES describes all mining model attributes in the database. Its columns are the same as those in ALL_MINING_MODEL_ATTRIBUTES.

See Also: "ALL_MINING_MODEL_ATTRIBUTES" on page 2-85

DBA_MINING_MODEL_SETTINGS

DBA_MINING_MODEL_SETTINGS describes all mining model settings in the database. Its columns are the same as those in ALL_MINING_MODEL_SETTINGS.

See Also: "ALL_MINING_MODEL_SETTINGS" on page 2-86

DBA_MINING_MODELS

DBA_MINING_MODELS describes all mining models in the database. Its columns are the same as those in ALL_MINING_MODELS.

See Also: "ALL_MINING_MODELS" on page 2-86

DBA_MVIEW_AGGREGATES

DBA_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of all aggregated materialized view in the database. Its columns are the same as those in "ALL_MVIEW_AGGREGATES" on page 2-87.

DBA_MVIEW_ANALYSIS

DBA_MVIEW_ANALYSIS describes all materialized views in the database that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in "ALL_MVIEW_ANALYSIS" on page 2-88.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as SYSDATE or USER. This view also excludes materialized views that were created as snapshots prior to Oracle8*i* and that were never altered to enable query rewrite.

DBA_MVIEW_COMMENTS

DBA_MVIEW_COMMENTS displays comments on all materialized views in the database. Its columns are the same as those in ALL_MVIEW_COMMENTS.

See Also: "ALL_MVIEW_COMMENTS" on page 2-89

DBA_MVIEW_DETAIL_PARTITION

DBA_MVIEW_DETAIL_PARTITION displays freshness information for all materialized views in the database, with respect to a PCT detail partition. Its columns are the same as those in "ALL_MVIEW_DETAIL_PARTITION" on page 2-90.

DBA_MVIEW_DETAIL_RELATIONS

DBA_MVIEW_DETAIL_RELATIONS represents the named detail relations that are either in the FROM list of a materialized view, or that are indirectly referenced through views in the FROM list. Its columns are the same as those in "ALL_MVIEW_DETAIL_RELATIONS" on page 2-90.

DBA_MVIEW_DETAIL_SUBPARTITION

DBA_MVIEW_DETAIL_SUBPARTITION displays freshness information for all materialized views in the database, with respect to a PCT detail subpartition. Its columns are the same as those in "ALL_MVIEW_DETAIL_SUBPARTITION" on page 2-91.

DBA_MVIEW_JOINS

DBA_MVIEW_JOINS describes a join between two columns in the WHERE clause of a subquery that defines a materialized view. Its columns are the same as those in "ALL_MVIEW_JOINS" on page 2-91.

DBA_MVIEW_KEYS

DBA_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views in the database are based. Its columns are the same as those in "ALL_MVIEW_KEYS" on page 2-92.

DBA_MVIEW_LOG_FILTER_COLS

DBA_MVIEW_LOG_FILTER_COLS lists all columns (excluding primary key columns) being logged in the materialized view logs.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the master table being logged
NAME	VARCHAR2(30)	NOT NULL	Name of the master table being logged
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column being logged

DBA_MVIEW_LOGS

DBA_MVIEW_LOGS describes all materialized view logs in the database. Its columns are the same as those in ALL_MVIEW_LOGS.

See Also: "ALL_MVIEW_LOGS" on page 2-92

DBA_MVIEW_REFRESH_TIMES

`DBA_MVIEW_REFRESH_TIMES` describes refresh times of all materialized views in the database. Its columns are the same as those in `ALL_MVIEW_REFRESH_TIMES`.

See Also: "ALL_MVIEW_REFRESH_TIMES" on page 2-93

DBA_MVIEWS

`DBA_MVIEWS` describes all materialized views in the database. Its columns are the same as those in `ALL_MVIEWS`.

See Also: "ALL_MVIEWS" on page 2-94

5

Static Data Dictionary Views: DBA_NESTED_ TABLE_COLS to USER_XML_VIEWS

This chapter contains the static data dictionary views DBA_NESTED_TABLE_COLS through USER_XML_VIEWS.

DBA_NESTED_TABLE_COLS

DBA_NESTED_TABLE_COLS describes the columns of all nested tables in the database. Its columns are the same as those in "ALL_NESTED_TABLE_COLS" on page 2-97. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_NESTED_TABLES

DBA_NESTED_TABLES describes all nested tables contained in all tables in the database. Its columns are the same as those in "ALL_NESTED_TABLES" on page 2-99.

DBA_NETWORK_ACL_PRIVILEGES

DBA_NETWORK_ACL_PRIVILEGES describes the network privileges defined in all access control lists that are currently assigned to network hosts.

Column	Datatype	NULL	Description
ACL	VARCHAR2(4000)		Path of the access control list
ACLID	RAW(16)	NOT NULL	Object ID of the access control list
PRINCIPAL	VARCHAR2(4000)		Principal (database user or role) whom the privilege is granted to or denied from
PRIVILEGE	VARCHAR2(7)		Network privilege
IS_GRANT	VARCHAR2(5)		Indicates whether the privilege is granted (true) or denied (false)
INVERT	VARCHAR2(5)		Indicates whether the access control entry contains invert principal (true) or not (false)
START_DATE	TIMESTAMP(9) WITH TIME ZONE		Start date of the access control entry
END_DATE	TIMESTAMP(9) WITH TIME ZONE		End date of the access control entry

DBA_NETWORK_ACLS

DBA_NETWORK_ACLS describes the access control list assignments to network hosts.

Column	Datatype	NULL	Description
HOST	VARCHAR2(1000)	NOT NULL	Network host
LOWER_PORT	NUMBER(5)		Lower bound of the port range
UPPER_PORT	NUMBER(5)		Upper bound of the port range
ACL	VARCHAR2(4000)		Path of the access control list
ACLID	RAW(16)	NOT NULL	Object ID of the access control list

DBA_OBJ_AUDIT_OPTS

DBA_OBJ_AUDIT_OPTS describes auditing options on all objects.

Related View

USER_OBJ_AUDIT_OPTS describes auditing options on all objects owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_TYPE	VARCHAR2(9)		Type of the object
ALT	VARCHAR2(3)		Auditing ALTER WHENEVER SUCCESSFUL/UNSUCCESSFUL
AUD	VARCHAR2(3)		Auditing AUDIT WHENEVER SUCCESSFUL/UNSUCCESSFUL
COM	VARCHAR2(3)		Auditing COMMENT WHENEVER SUCCESSFUL/UNSUCCESSFUL
DEL	VARCHAR2(3)		Auditing DELETE WHENEVER SUCCESSFUL/UNSUCCESSFUL
GRA	VARCHAR2(3)		Auditing GRANT WHENEVER SUCCESSFUL/UNSUCCESSFUL
IND	VARCHAR2(3)		Auditing INDEX WHENEVER SUCCESSFUL/UNSUCCESSFUL
INS	VARCHAR2(3)		Auditing INSERT WHENEVER SUCCESSFUL/UNSUCCESSFUL
LOC	VARCHAR2(3)		Auditing LOCK WHENEVER SUCCESSFUL/UNSUCCESSFUL
REN	VARCHAR2(3)		Auditing RENAME WHENEVER SUCCESSFUL/UNSUCCESSFUL
SEL	VARCHAR2(3)		Auditing SELECT WHENEVER SUCCESSFUL/UNSUCCESSFUL
UPD	VARCHAR2(3)		Auditing UPDATE WHENEVER SUCCESSFUL/UNSUCCESSFUL
REF	CHAR(3)		Auditing REFERENCE WHENEVER SUCCESSFUL/UNSUCCESSFUL (not used)
EXE	VARCHAR2(3)		Auditing EXECUTE WHENEVER SUCCESSFUL/UNSUCCESSFUL
CRE	VARCHAR2(3)		Auditing CREATE WHENEVER SUCCESSFUL / UNSUCCESSFUL
REA	VARCHAR2(3)		Auditing READ WHENEVER SUCCESSFUL/UNSUCCESSFUL
WRI	VARCHAR2(3)		Auditing WRITE WHENEVER SUCCESSFUL / UNSUCCESSFUL

Column	Datatype	NULL	Description
FBK	VARCHAR2(3)		Auditing FLASHBACK WHENEVER SUCCESSFUL / UNSUCCESSFUL

DBA_OBJ_COLATTRS

DBA_OBJ_COLATTRS describes object columns and attributes contained in all tables in the database. Its columns are the same as those in ALL_OBJ_COLATTRS.

See Also: "ALL_OBJ_COLATTRS" on page 2-99

DBA_OBJECT_SIZE

DBA_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects.

Related View

- USER_OBJECT_SIZE lists the size of PL/SQL objects owned by the current user.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
TYPE	VARCHAR2(12)		Type of the object: TYPE, TYPE BODY, TABLE, VIEW, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, JAVA SOURCE, JAVA CLASS or JAVA RESOURCE
SOURCE_SIZE	NUMBER		Size of the source in bytes. Must be in memory during compilation, or dynamic recompilation.
PARSED_SIZE	NUMBER		Size of the parsed form of the object, in bytes. Must be in memory when an object is being compiled that references this object.
CODE_SIZE	NUMBER		Code size, in bytes. Must be in memory when this object is executing.
ERROR_SIZE	NUMBER		Size of error messages, in bytes. In memory during the compilation of the object when there are compilation errors.

DBA_OBJECT_TABLES

DBA_OBJECT_TABLES describes all object tables in the database. Its columns are the same as those in ALL_OBJECT_TABLES.

See Also: "ALL_OBJECT_TABLES" on page 2-100

DBA_OBJECTS

DBA_OBJECTS describes all objects in the database. Its columns are the same as those in "ALL_OBJECTS" on page 2-102.

DBA_OPANCILLARY

DBA_OPANCILLARY provides ancillary information for all operator bindings in the database. Its columns are the same as those in "ALL_OPANCILLARY" on page 2-103.

DBA_OPARGUMENTS

DBA_OPARGUMENTS provides argument information for all operator bindings in the database. Its columns are the same as those in "ALL_OPARGUMENTS" on page 2-103.

DBA_OPBINDINGS

DBA_OPBINDINGS describes the binding functions and methods on all operators in the database. Its columns are the same as those in ALL_OPBINDINGS.

See Also: "ALL_OPBINDINGS" on page 2-103

DBA_OPERATOR_COMMENTS

DBA_OPERATOR_COMMENTS displays comments for all user-defined operators in the database. Its columns are the same as those in ALL_OPERATOR_COMMENTS.

See Also: "ALL_OPERATOR_COMMENTS" on page 2-104

DBA_OPERATORS

DBA_OPERATORS describes all operators in the database. Its columns are the same as those in ALL_OPERATORS.

See Also: "ALL_OPERATORS" on page 2-105

DBA_OPTSTAT_OPERATIONS

DBA_OPTSTAT_OPERATIONS contains a history of statistics operations performed at the schema and database level using the DBMS_STATS package.

Column	Datatype	NULL	Description
OPERATION	VARCHAR2(64)		Operation name
TARGET	VARCHAR2(64)		Target on which the operation was performed
START_TIME	TIMESTAMP(6) WITH TIME ZONE		Time at which the operation started
END_TIME	TIMESTAMP(6) WITH TIME ZONE		Time at which the operation ended

DBA_ORPHAN_KEY_TABLE

DBA_ORPHAN_KEY_TABLE reports key values from indexes where the underlying base table has block corruptions. To create the view, run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the orphan key table for an index, run the DBMS_REPAIR.DUMP_ORPHAN_KEYS procedure on the index. For each key in the index that points to a corrupt data block, Oracle inserts a row into the orphan key table.

Column	Datatype	NULL	Description
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name of the index
INDEX_NAME	VARCHAR2(30)	NOT NULL	Name of the index
IPART_NAME	VARCHAR2(30)	NULL	Name of the index partition or subpartition
INDEX_ID	NUMBER	NOT NULL	Dictionary object ID of the index
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the base table of the index

Column	Datatype	NULL	Description
PART_NAME	VARCHAR2(30)	NULL	Name of the base table partition or subpartition
TABLE_ID	NUMBER	NOT NULL	Dictionary object ID of the base table
KEYDATA	UROWID	NOT NULL	Physical rowid of the corrupt data row
KEY	UROWID	NOT NULL	Key values for the index entry
DUMP_TIMESTAMP	DATE	NOT NULL	Timestamp when the entry was made into the orphan key table

DBA_OUTLINE_HINTS

DBA_OUTLINE_HINTS describes the set of hints stored in all outlines in the database.

Related View

USER_OUTLINE_HINTS describes the set of hints stored in the outlines owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the outline
OWNER	VARCHAR2(30)		Name of the user who created the outline
NODE	NUMBER		ID of the query or subquery to which the hint applies. The top-level query is labeled 1. Subqueries are assigned sequentially numbered labels, starting with 2.
STAGE	NUMBER		Outline hints can be applied at three different stages during the compilation process. This column indicates the stage at which this hint was applied.
JOIN_POS	NUMBER		Position of the table in the join order. The value is 0 for all hints except access method hints, which identify a table to which the hint and the join position apply.
HINT	VARCHAR2(512)		Text of the hint

See Also: "USER_OUTLINE_HINTS" on page 5-120

DBA_OUTLINES

DBA_OUTLINES describes all stored outlines in the database.

Related View

USER_OUTLINES describes the stored outlines owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		User-specified or generated name of the stored outline. The name must be of a form that can be expressed in SQL.
OWNER	VARCHAR2(30)		Name of the user who created the outline
CATEGORY	VARCHAR2(30)		User-defined name of the category to which the outline belongs
USED	VARCHAR2(6)		Indicates whether the outline has ever been used (USED) or not (UNUSED)
TIMESTAMP	DATE		Timestamp of outline creation
VERSION	VARCHAR2(64)		Oracle version that created the outline

Column	Datatype	NULL	Description
SQL_TEXT	LONG		SQL text of the query, including any hints that were a part of the original statement. If bind variables are included, the variable names are stored as SQL text, not the values that are assigned to the variables. **Note:** This field may contain sensitive information about your database or application. Therefore, use discretion when granting SELECT or VIEW object privileges on these views.
SIGNATURE	RAW(16)		Signature uniquely identifying the outline SQL text
COMPATIBLE	VARCHAR2(12)		Indicates whether the outline hints were compatible across a migration (COMPATIBLE) or not (INCOMPATIBLE)
ENABLED	VARCHAR2(8)		Indicates whether the outline is enabled (ENABLED) or disabled (DISABLED)
FORMAT	VARCHAR2(6)		Hint format: ■ NORMAL ■ LOCAL

See Also: "USER_OUTLINES" on page 5-120

DBA_OUTSTANDING_ALERTS

DBA_OUTSTANDING_ALERTS describes alerts which the server considers to be outstanding.

Column	Datatype	NULL	Description
SEQUENCE_ID	NUMBER		Alert sequence number
REASON_ID	NUMBER	NOT NULL	ID of the alert reason
OWNER	VARCHAR2(30)		Owner of the object on which the alert was issued
OBJECT_NAME	VARCHAR2(513)		Name of the object
SUBOBJECT_NAME	VARCHAR2(30)		Name of the subobject
OBJECT_TYPE	VARCHAR2(64)		Object type
REASON	VARCHAR2(4000)		Reason for the alert
TIME_SUGGESTED	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was last updated
CREATION_TIME	TIMESTAMP(6) WITH TIME ZONE		Time when the alert was first created
SUGGESTED_ACTION	VARCHAR2(4000)		Advice of the recommended action
ADVISOR_NAME	VARCHAR2(30)		Name of the advisor to be invoked for more information
METRIC_VALUE	NUMBER		Value of the related metrics
MESSAGE_TYPE	VARCHAR2(12)		Message type: ■ Notification ■ Warning
MESSAGE_GROUP	VARCHAR2(64)		Name of the message group to which the alert belongs
MESSAGE_LEVEL	NUMBER		Message severity level (1 to 32)
HOSTING_CLIENT_ID	VARCHAR2(64)		ID of the client or security group to which the alert relates
MODULE_ID	VARCHAR2(64)		ID of the module that originated the alert
PROCESS_ID	VARCHAR2(128)		Process ID

Column	Datatype	NULL	Description
HOST_ID	VARCHAR2(256)		DNS host name of the originating host
HOST_NW_ADDR	VARCHAR2(256)		IP or other network address of the originating host
INSTANCE_NAME	VARCHAR2(16)		Originating instance name
INSTANCE_NUMBER	NUMBER		Originating instance number
USER_ID	VARCHAR2(30)		User ID
EXECUTION_CONTEXT_ID	VARCHAR2(60)		ID of the threshold of execution
ERROR_INSTANCE_ID	VARCHAR2(142)		ID of an error instance plus a sequence number

DBA_PART_COL_STATISTICS

DBA_PART_COL_STATISTICS displays column statistics and histogram information for all table partitions in the database. Its columns are the same as those in ALL_PART_COL_STATISTICS.

See Also: "ALL_PART_COL_STATISTICS" on page 3-1

DBA_PART_HISTOGRAMS

DBA_PART_HISTOGRAMS displays the histogram data (endpoints per histogram) for the histograms on all table partitions in the database. Its columns are the same as those in ALL_PART_HISTOGRAMS.

See Also: "ALL_PART_HISTOGRAMS" on page 3-2

DBA_PART_INDEXES

DBA_PART_INDEXES displays the object-level partitioning information for all partitioned indexes in the database. Its columns are the same as those in ALL_PART_INDEXES.

See Also: "ALL_PART_INDEXES" on page 3-2

DBA_PART_KEY_COLUMNS

DBA_PART_KEY_COLUMNS describes the partitioning key columns for all partitioned objects in the database. Its columns are the same as those in ALL_PART_KEY_COLUMNS.

See Also: "ALL_PART_KEY_COLUMNS" on page 3-4

DBA_PART_LOBS

DBA_PART_LOBS displays table-level information for all partitioned LOBs in the database, including default attributes for LOB data partitions. Its columns are the same as those in ALL_PART_LOBS.

See Also: "ALL_PART_LOBS" on page 3-5

DBA_PART_TABLES

DBA_PART_TABLES displays the object-level partitioning information for all partitioned tables in the database. Its columns are the same as those in ALL_PART_TABLES.

See Also: "ALL_PART_TABLES" on page 3-7

DBA_PARTIAL_DROP_TABS

DBA_PARTIAL_DROP_TABS describes all tables in the database that have partially completed DROP COLUMN operations. Its columns are the same as those in "ALL_PARTIAL_DROP_TABS" on page 3-9.

DBA_PENDING_CONV_TABLES

DBA_PENDING_CONV_TABLES describes all pending conversion tables in the database. Its columns are the same as those in ALL_PENDING_CONV_TABLES.

See Also: "ALL_PENDING_CONV_TABLES" on page 3-9

DBA_PENDING_TRANSACTIONS

DBA_PENDING_TRANSACTIONS describes unresolved transactions (either due to failure or if the coordinator has not sent a commit/rollback).

Column	Datatype	NULL	Description
FORMATID	NUMBER		The format identifier of the transaction identifier
GLOBALID	RAW(64)		The global part (gtrid) of the transaction identifier
BRANCHID	RAW(64)		The branch qualifier (bqual) of the transaction identifier

DBA_PLSQL_OBJECT_SETTINGS

DBA_PLSQL_OBJECT_SETTINGS displays information about the compiler settings for all stored objects in the database. Its columns are the same as those in ALL_PLSQL_OBJECT_SETTINGS.

See Also: "ALL_PLSQL_OBJECT_SETTINGS" on page 3-10

DBA_POLICIES

DBA_POLICIES describes all Oracle Virtual Private Database (VPD) security policies in the database. Its columns are the same as those in ALL_POLICIES. A security policy is a list of security requirements and rules that regulate user access to database objects.

See Also: "ALL_POLICIES" on page 3-10

DBA_POLICY_CONTEXTS

DBA_POLICY_CONTEXTS describes all driving contexts in the database. Its columns are the same as those in ALL_POLICY_CONTEXTS.

See Also: "ALL_POLICY_CONTEXTS" on page 3-12

DBA_POLICY_GROUPS

DBA_POLICY_GROUPS describes all policy groups in the database. Its columns are the same as those in ALL_POLICY_GROUPS.

See Also: "ALL_POLICY_GROUPS" on page 3-12

DBA_PRIV_AUDIT_OPTS

DBA_PRIV_AUDIT_OPTS describes current system privileges being audited across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited. NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client; NULL if the client is performing the operation directly.
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege being audited
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_PROCEDURES

DBA_PROCEDURES lists all functions and procedures along with their associated properties. Its columns are the same as those in "ALL_PROCEDURES" on page 3-12.

DBA_PROFILES

DBA_PROFILES displays all profiles and their limits.

Column	Datatype	NULL	Description
PROFILE	VARCHAR2(30)	NOT NULL	Profile name
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Resource name
RESOURCE_TYPE	VARCHAR2(8)		Indicates whether the resource profile is a KERNEL or a PASSWORD parameter
LIMIT	VARCHAR2(40)		Limit placed on this resource for this profile

DBA_PROPAGATION

DBA_PROPAGATION displays information about all Streams propagations in the database. Its columns are the same as those in ALL_PROPAGATION.

See Also: "ALL_PROPAGATION" on page 3-13

DBA_PROXIES

DBA_PROXIES displays Information about all proxy connections in the database.

Related View

USER_PROXIES displays information about connections the current user is allowed to proxy. This view does not display the PROXY or PROXY_AUTHORITY columns.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of the proxy user
CLIENT	VARCHAR2(30)	NOT NULL	Name of the client user who the proxy user can act on behalf of
AUTHENTICATION	VARCHAR2(3)		Indicates whether the proxy is required to supply the client's authentication credentials (YES) or not (NO)
AUTHORIZATION_CONSTRAINT	VARCHAR2(35)		Indicates the proxy's authority to exercise roles on the client's behalf: ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ NO CLIENT ROLES MAY BE ACTIVATED ■ PROXY MAY ACTIVATE ROLE ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ PROXY MAY NOT ACTIVATE ROLE
ROLE	VARCHAR2(30)		Name of the role referenced in AUTHORIZATION_CONSTRAINT
PROXY_AUTHORITY	VARCHAR2(9)		Value is either: ■ DIRECTORY if EUS proxy is enabled for that database user ■ DATABASE if this row describes a local proxy permission

See Also: "USER_PROXIES" on page 5-122

DBA_PUBLISHED_COLUMNS

DBA_PUBLISHED_COLUMNS describes all source table columns in the database which have been published for the source tables. This view is intended for use by Change Data Capture publishers.

Related View

USER_PUBLISHED_COLUMNS describes the source table columns owned by the current user which have been published for the source tables. This view does not display the CHANGE_TABLE_SCHEMA or CHANGE_TABLE_NAME columns.

Column	Datatype	NULL	Description
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Name of the change set
CHANGE_TABLE_SCHEMA	VARCHAR2(30)	NOT NULL	Associated change table owner
CHANGE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Associated change table name
PUB_ID	NUMBER	NOT NULL	Publication identifier
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Table owner in the source database
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name in the source database
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Column name
DATA_TYPE	VARCHAR2(106)		Column datatype
DATA_LENGTH	NUMBER	NOT NULL	Column length (in bytes)
DATA_PRECISION	NUMBER		Decimal precision for NUMBER datatypes; binary precision for FLOAT datatypes; NULL for all other datatypes

Column	Datatype	NULL	Description
DATA_SCALE	NUMBER		Digits to the right of a decimal point in a number
NULLABLE	VARCHAR2(1)		Indicates whether nulls are allowed (Y) or not (N)

See Also: "USER_PUBLISHED_COLUMNS" on page 5-122

DBA_QUEUE_SCHEDULES

DBA_QUEUE_SCHEDULES describes all propagation schedules in the database.

Related View

USER_QUEUE_SCHEDULES describes the propagation schedules whose source queues are owned by the current user. This view does not display the SCHEMA column.

Column	Datatype	NULL	Description
SCHEMA	VARCHAR2(30)		Source queue owner
QNAME	VARCHAR2(30)		Source queue name
DESTINATION	VARCHAR2(128)		Destination name, currently limited to be a DBLINK name
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Date at which to start propagation
START_TIME	VARCHAR2(8)		Time of day at which to start propagation (in HH:MI:SS format)
PROPAGATION_WINDOW	NUMBER		Duration for the propagation window (in seconds)
NEXT_TIME	VARCHAR2(4000)		Function to compute the start of the next propagation window
LATENCY	NUMBER		Maximum wait time to propagate a message during the propagation window
SCHEDULE_DISABLED	VARCHAR2(1)		Indicates whether the schedule is disabled (Y) or enabled (N). If disabled, then the schedule will not be executed.
PROCESS_NAME	VARCHAR2(4)		Name of the process executing the schedule; NULL if not currently executing
SESSION_ID	VARCHAR2(82)		Session ID and session serial number of the job executing this schedule (SID, SERIAL#); NULL if not currently executing
INSTANCE	NUMBER		Cluster database instance number executing the schedule
LAST_RUN_DATE	TIMESTAMP(6) WITH TIME ZONE		Date of the last successful execution
LAST_RUN_TIME	VARCHAR2(8)		Time of day of the last successful execution (in HH:MI:SS format)
CURRENT_START_DATE	DATE		Date at which the current window of this schedule was started
CURRENT_START_TIME	VARCHAR2(8)		Time of day at which the current window of this schedule was started (in HH:MI:SS format)
NEXT_RUN_DATE	TIMESTAMP(6) WITH TIME ZONE		Date at which the next window of this schedule will be started
NEXT_RUN_TIME	VARCHAR2(8)		Time of day at which the next window of this schedule will be started (in HH:MI:SS format)
TOTAL_TIME	NUMBER		Total time spent by the system in executing this schedule (in seconds)

Column	Datatype	NULL	Description
TOTAL_NUMBER	NUMBER		Total number of messages propagated in this schedule
TOTAL_BYTES	NUMBER		Total number of bytes propagated in this schedule
MAX_NUMBER	NUMBER		Maximum number of messages propagated in a propagation window
MAX_BYTES	NUMBER		Maximum number of bytes propagated in a propagation window
AVG_NUMBER	NUMBER		Average number of messages propagated in a propagation window
AVG_SIZE	NUMBER		Average size of a propagated message (in bytes)
AVG_TIME	NUMBER		Average time to propagate a message (in seconds)
FAILURES	NUMBER		Number of consecutive times schedule execution has failed, if any. After 16 consecutive failures, a propagation job becomes disabled automatically.
LAST_ERROR_DATE	DATE		Date of the last unsuccessful execution
LAST_ERROR_TIME	VARCHAR2(8)		Time of day of the last unsuccessful execution (in HH:MI:SS format)
LAST_ERROR_MSG	VARCHAR2(4000)		Error number and error message text of the last unsuccessful execution
MESSAGE_DELIVERY_MODE	VARCHAR2(10)		Message delivery mode: ■ PERSISTENT ■ BUFFERED
ELAPSED_DEQUEUE_TIME	NUMBER		Elapsed dequeue time (in hundredths of a second)
ELAPSED_PICKLE_TIME	NUMBER		Elapsed pickle time (time taken to linearize a logical change record (LCR) into a stream of bytes that can be sent over the network) (in hundredths of a second)
JOB_NAME	VARCHAR2(30)		Name of the Scheduler job

See Also: "USER_QUEUE_SCHEDULES" on page 5-122

DBA_QUEUE_SUBSCRIBERS

DBA_QUEUE_SUBSCRIBERS returns all subscribers on all queues in the

database. Its columns are the same as those in "ALL_QUEUE_SUBSCRIBERS" on page 3-14.

DBA_QUEUE_TABLES

DBA_QUEUE_TABLES contains information about the owner instance for a queue table. A queue table can contain multiple queues. In this case, each queue in a queue table has the same owner instance as the queue table. Its columns are the same as those in ALL_QUEUE_TABLES.

See Also: "ALL_QUEUE_TABLES" on page 3-15

DBA_QUEUES

DBA_QUEUES describes the operational characteristics of every queue in a database. Its columns are the same as those in "ALL_QUEUES" on page 3-16.

See Also: *Oracle Streams Advanced Queuing User's Guide* for more information about this view and Advanced Queuing

DBA_RCHILD

DBA_RCHILD lists all the children in any refresh group.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group
TYPE#	VARCHAR2(30)		Type of the object in the refresh group

DBA_RECOVERABLE_SCRIPT

DBA_RECOVERABLE_SCRIPT provides details about recoverable operations.

Column	Datatype	NULL	Description
SCRIPT_ID	RAW(16)		Unique ID of the operation
CREATION_TIME	DATE		Time the operation was invoked
INVOKING_PACKAGE_OWNER	VARCHAR2(30)		Invoking package owner of the operation
INVOKING_PACKAGE	VARCHAR2(30)		Invoking package of the operation
INVOKING_PROCEDURE	VARCHAR2(30)		Invoking procedure of the operation
INVOKING_USER	VARCHAR2(30)		Script owner
STATUS	VARCHAR2(12)		State of the recoverable script: GENERATING, NOT EXECUTED, EXECUTING, EXECUTED, or ERROR
TOTAL_BLOCKS	NUMBER		Total number of blocks for the recoverable script to be executed
DONE_BLOCK_NUM	NUMBER		Last block executed, thus far
SCRIPT_COMMENT	VARCHAR2(4000)		Comment for the recoverable script

DBA_RECOVERABLE_SCRIPT_BLOCKS

DBA_RECOVERABLE_SCRIPT_BLOCKS provides details about recoverable script blocks.

Column	Datatype	NULL	Description
SCRIPT_ID	RAW(16)		Global unique ID of the recoverable script to which this block belongs
BLOCK_NUM	NUMBER		The *n*th block in the recoverable script to be executed
FORWARD_BLOCK	CLOB		Forward block to be executed
FORWARD_BLOCK_DBLINK	VARCHAR2(128)		Database where the forward block is executed
UNDO_BLOCK	CLOB		Block to roll back the forward operation
UNDO_BLOCK_DBLINK	VARCHAR2(128)		Database where the undo block is executed
STATUS	VARCHAR2(12)		Status of the block execution: GENERATING, NOT EXECUTED, EXECUTING, EXECUTED, or ERROR
BLOCK_COMMENT	VARCHAR2(4000)		Comment for the block

DBA_RECOVERABLE_SCRIPT_ERRORS

DBA_RECOVERABLE_SCRIPT_ERRORS provides details about errors that occurred during script execution.

Column	Datatype	NULL	Description
SCRIPT_ID	RAW(16)		Global unique ID of the recoverable script
BLOCK_NUM	NUMBER		The *n*th block that failed
ERROR_NUMBER	NUMBER		Number of the error encountered while executing the block
ERROR_MESSAGE	VARCHAR2(4000)		Error message encountered while executing the block
ERROR_CREATION_TIME	DATE		Time that the error was created

DBA_RECOVERABLE_SCRIPT_PARAMS

DBA_RECOVERABLE_SCRIPT_PARAMS provides details about recoverable operation parameters.

Column	Datatype	NULL	Description
SCRIPT_ID	RAW(16)		Unique ID of the operation
PARAMETER	VARCHAR2(30)		Name of the parameter
PARAM_INDEX	NUMBER		Index for multi-valued parameter
VALUE	VARCHAR2(4000)		Value of the parameter

DBA_RECYCLEBIN

DBA_RECYCLEBIN displays information about all recycle bins in the database.

Related View

USER_RECYCLEBIN displays information about the recycle bin owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Name of the original owner of the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	New name of the object
ORIGINAL_NAME	VARCHAR2(32)		Original name of the object
OPERATION	VARCHAR2(9)		Operation carried out on the object: ■ DROP - Object was dropped ■ TRUNCATE - Object was truncated **Note:** The Oracle Database currently only supports recovering dropped objects from the recycle bin. The truncated objects cannot be recovered.

Column	Datatype	NULL	Description
TYPE	VARCHAR2(25)		Type of the object: - TABLE - NORMAL INDEX - BITMAP INDEX - NESTED TABLE - LOB - LOB INDEX - DOMAIN INDEX - IOT TOP INDEX - IOT OVERFLOW SEGMENT - IOT MAPPING TABLE - TRIGGER - CONSTRAINT - Table Partition - Table Composite Partition - Index Partition - Index Composite Partition - LOB Partition - LOB Composite Partition
TS_NAME	VARCHAR2(30)		Name of the tablespace to which the object belongs
CREATETIME	VARCHAR2(19)		Timestamp for the creation of the object
DROPTIME	VARCHAR2(19)		Timestamp for the dropping of the object
DROPSCN	NUMBER		System change number (SCN) of the transaction which moved the object to the recycle bin
PARTITION_NAME	VARCHAR2(32)		Name of the partition which was dropped
CAN_UNDROP	VARCHAR2(3)		Indicates whether the object can be undropped (YES) or not (NO)
CAN_PURGE	VARCHAR2(3)		Indicates whether the object can be purged (YES) or not (NO)
RELATED	NUMBER	NOT NULL	Object number of the parent object
BASE_OBJECT	NUMBER	NOT NULL	Object number of the base object
PURGE_OBJECT	NUMBER	NOT NULL	Object number for the object which gets purged
SPACE	NUMBER		Number of blocks used by the object

See Also: "USER_RECYCLEBIN" on page 5-123

DBA_REDEFINITION_ERRORS

DBA_REDEFINITION_ERRORS is an online redefinition view and displays the dependent objects for which errors were raised while attempting to create similar objects on the interim table of the redefinition.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(10)		Type of the redefinition object: - TABLE - INDEX - CONSTRAINT - TRIGGER

Column	Datatype	NULL	Description
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the redefinition object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the redefinition object
BASE_TABLE_OWNER	VARCHAR2(30)		Owner of the base table of the redefinition object
BASE_TABLE_NAME	VARCHAR2(30)		Name of the base table of the redefinition object
DDL_TXT	CLOB		DDL used to create the corresponding interim redefinition object

DBA_REDEFINITION_OBJECTS

DBA_REDEFINITION_OBJECTS is an online redefinition view and displays the objects involved in the current redefinitions.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(12)		Type of the redefinition object: ■ TABLE ■ INDEX ■ CONSTRAINT ■ TRIGGER ■ NESTED TABLE ■ PARTITION
OBJECT_OWNER	VARCHAR2(30)	NOT NULL	Owner of the redefinition object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the redefinition object
BASE_TABLE_OWNER	VARCHAR2(30)		Owner of the base table of the redefinition object
BASE_TABLE_NAME	VARCHAR2(30)		Name of the base table of the redefinition object
INTERIM_OBJECT_OWNER	VARCHAR2(30)		Owner of the corresponding interim redefinition object
INTERIM_OBJECT_NAME	VARCHAR2(30)		Name of the corresponding interim redefinition object

DBA_REFRESH

DBA_REFRESH describes all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH" on page 3-16.

DBA_REFRESH_CHILDREN

DBA_REFRESH_CHILDREN lists all of the objects in all refresh groups in the database. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 3-17.

DBA_REFS

DBA_REFS describes the REF columns and REF attributes in object type columns of all the objects in the database. Its columns are the same as those in "ALL_REFS" on page 3-18.

DBA_REGISTERED_ARCHIVED_LOG

DBA_REGISTERED_ARCHIVED_LOG displays information about all registered archived logfiles in the database.

Column	Datatype	NULL	Description
CONSUMER_NAME	VARCHAR2(30)	NOT NULL	Consumer name of the archived logs
SOURCE_DATABASE	VARCHAR2(128)		Name of the database which generated the redo logs
THREAD#	NUMBER	NOT NULL	Thread number of the archived redo log. The thread number is 1 for a single instance. For Real Application Clusters, this column will contain different numbers.
SEQUENCE#	NUMBER	NOT NULL	Sequence number of the archived redo log file
FIRST_SCN	NUMBER	NOT NULL	System change number (SCN) of the current archived redo log
NEXT_SCN	NUMBER		System change number (SCN) of the next archived redo log
FIRST_TIME	DATE		Date and time of the current archived redo log
NEXT_TIME	DATE		Date and time of the next archived redo log
NAME	VARCHAR2(513)		Name of the archived redo log
MODIFIED_TIME	DATE		Time when the archived redo log was registered
DICTIONARY_BEGIN	VARCHAR2(3)		Indicates whether the beginning of the dictionary build is in the archived redo log (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)		Indicates whether the end of the dictionary build is in the archived redo log (YES) or not (NO)
PURGEABLE	VARCHAR2(3)		Indicates whether the redo log can be permanently removed (YES) or not (NO)
RESET_LOGS_CHANGE#	NUMBER	NOT NULL	Resetlogs change number of the database when the log was written
RESET_TIMESTAMP	NUMBER	NOT NULL	Resetlogs time of the database when the log was written

DBA_REGISTERED_MVIEW_GROUPS

DBA_REGISTERED_MVIEW_GROUPS lists all the registered materialized view groups at the master site or master materialized view site.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the materialized view replication group
MVIEW_SITE	VARCHAR2(128)		Site of the materialized view replication group
GROUP_COMMENT	VARCHAR2(80)		Description of the materialized view replication group
VERSION	VARCHAR2(8)		Oracle release of the materialized view replication group: ■ ORACLE 7 ■ ORACLE 8 ■ REPAPI **Note:** Oracle8*i* and newer materialized view groups show ORACLE 8.
FNAME	VARCHAR2(30)		Name of the flavor of the materialized view group
OWNER	VARCHAR2(30)		Owner of the materialized view replication group

DBA_REGISTERED_MVIEWS

DBA_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) in the database. Its columns are the same as those in ALL_REGISTERED_MVIEWS.

See Also: "ALL_REGISTERED_MVIEWS" on page 3-19

DBA_REGISTRY

DBA_REGISTRY displays information about the components loaded into the database.

Related View

USER_REGISTRY displays information about the components loaded into the database that are owned by the current user.

Column	Datatype	NULL	Description
COMP_ID	VARCHAR2(30)	NOT NULL	Component identifier
COMP_NAME	VARCHAR2(255)		Component name
VERSION	VARCHAR2(30)		Component version loaded
STATUS	VARCHAR2(11)		Component status: ▪ INVALID ▪ VALID ▪ LOADING ▪ LOADED ▪ UPGRADING ▪ UPGRADED ▪ DOWNGRADING ▪ DOWNGRADED ▪ REMOVING ▪ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified
NAMESPACE	VARCHAR2(30)	NOT NULL	Component namespace
CONTROL	VARCHAR2(30)	NOT NULL	User that created the component entry
SCHEMA	VARCHAR2(30)	NOT NULL	User that contains the objects for the component
PROCEDURE	VARCHAR2(61)		Validation procedure
STARTUP	VARCHAR2(8)		Indicates whether the component requires a startup after the upgrade (REQUIRED) or not
PARENT_ID	VARCHAR2(30)		Parent component identifier

See Also: "USER_REGISTRY" on page 5-123

DBA_REGISTRY_HIERARCHY

DBA_REGISTRY_HIERARCHY displays information about the components loaded into the database, grouped by owner and organized in the component hierarchy.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)	NOT NULL	Component namespace
COMP_ID	VARCHAR2(4000)		Component identifier
VERSION	VARCHAR2(30)		Component version loaded

Column	Datatype	NULL	Description
STATUS	VARCHAR2(11)		Component status: ■ INVALID ■ VALID ■ LOADING ■ LOADED ■ UPGRADING ■ UPGRADED ■ DOWNGRADING ■ DOWNGRADED ■ REMOVING ■ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified

DBA_REGISTRY_HISTORY

DBA_REGISTRY_HISTORY provides information about upgrades, downgrades, and critical patch updates that have been performed on the database.

Column	Datatype	NULL	Description
ACTION_TIME	TIMESTAMP(6)		The time the upgrade, downgrade, or patch action was completed
ACTION	VARCHAR2(30)		The specific action (for example, UPGRADE or DOWNGRADE)
NAMESPACE	VARCHAR2(30)		The namespace of the components affected (for example, SERVER)
VERSION	VARCHAR2(30)		The version number of the server (for example, 10.2.0.1.0)
ID	NUMBER		The identification number of the Critical Patch Update
COMMENTS	VARCHAR2(255)		Additional comments about the action taken

DBA_REGISTRY_LOG

DBA_REGISTRY_LOG displays operating information about components loaded into the database.

Column	Datatype	NULL	Description
OPTIME	VARCHAR2(20)		Operation time
NAMESPACE	VARCHAR2(30)		Component namespace
COMP_ID	VARCHAR2(30)		Component identifier
OPERATION	VARCHAR2(11)		Operation name
MESSAGE	VARCHAR2(1000)		Message

DBA_REPAIR_TABLE

DBA_REPAIR_TABLE describes any corruptions found by the DBMS_REPAIR.CHECK_OBJECT procedure. This information is used by the DBMS_REPAIR.FIX_CORRUPT_BLOCKS procedure on execution. To create this view, first run the DBMS_REPAIR.ADMIN_TABLES procedure. To populate the resulting repair table for an object, run the DBMS_REPAIR.CHECK_OBJECT procedure on the object.

Note: The table created by the DBMS_REPAIR.ADMIN_TABLES procedure is called REPAIR TABLE by default. If you specify a different name, this view will have the name you specify, preceded by "DBA_REPAIR_".

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Dictionary object number of the object with the corruption
TABLESPACE_ID	NUMBER	NOT NULL	Tablespace number of the corrupt object
RELATIVE_FILE_ID	NUMBER)	NOT NULL	Relative file number of the corrupt object
BLOCK_ID	NUMBER	NOT NULL	Block number of the corruption
CORRUPT_TYPE	NUMBER	NOT NULL	Type of corruption encountered
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema of the corrupt object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the corrupt object
BASEOBJECT_NAME	VARCHAR2(30)	NULL	If the object is an index, the name of its base table
PARTITION_NAME	VARCHAR2(30)	NULL	Partition or subpartition name, if applicable
CORRUPT_DESCRIPTION	VARCHAR2(200)	NULL	Description of corruption
REPAIR_DESCRIPTION	VARCHAR2(200)	NULL	Description of repair action
MARKED_CORRUPT	VARCHAR2(10)	NOT NULL	Whether the block is marked corrupt (TRUE | FALSE)
CHECK_TIMESTAMP	DATE	NOT NULL	Date and time when this row was insert into the repair table
FIX_TIMESTAMP	DATE	NULL	Date and time when the block was modified by the FIX_CORRUPT_BLOCKS procedure, if applicable
REFORMAT_TIMESTAMP	DATE	NULL	Reserved for future use

DBA_RESOURCE_INCARNATIONS

DBA_RESOURCE_INCARNATIONS lists all resource incarnations that are running or eligible for HA status notification.

Column	Datatype	NULL	Description
RESOURCE_TYPE	VARCHAR2(30)	NOT NULL	Type of resource
RESOURCE_NAME	VARCHAR2(256)		Name of resource
DB_UNIQUE_NAME	VARCHAR2(30)	NOT NULL	Database unique name
DB_DOMAIN	VARCHAR2(128)	NOT NULL	Database domain
INSTANCE_NAME	VARCHAR2(30)	NOT NULL	Name of instance at which resource is located
HOST_NAME	VARCHAR2(512)		Name of host at which resource is located
STARTUP_TIME	TIMESTAMP(9) WITH TIME ZONE		Resource startup date and time

DBA_RESUMABLE

DBA_RESUMABLE lists all resumable statements executed in the system.

Related View

USER_RESUMABLE lists resumable statements executed by the current user. This view does not display the USER_ID column.

Column	Datatype	NULL	Description
USER_ID	NUMBER		User ID Number of the Resumable Statement Owner
SESSION_ID	NUMBER		Session Identifier of the Resumable Statement
INSTANCE_ID	NUMBER		Instance Number of the Resumable Statement
COORD_INSTANCE_ID	NUMBER		Instance Number on which the Parallel Coordinator is Running
COORD_SESSION_ID	NUMBER		Session Identifier of the Parallel Coordinator
STATUS	VARCHAR2(9)		Status of the resumable statement: ■ RUNNING ■ SUSPENDED ■ TIMEOUT ■ ERROR ■ ABORTED
TIMEOUT	NUMBER		Timeout of the resumable statement
START_TIME	VARCHAR2(20)		Start time of the resumable statement
SUSPEND_TIME	VARCHAR2(20)		Last time the resumable statement was suspended. It is initialized to null.
RESUME_TIME	VARCHAR2(20)		Last time the suspended resumable statement was resumed. It is initialized to null.
NAME	VARCHAR2(4000)		Name given in the resumable clause of the resumable statement
SQL_TEXT	VARCHAR2(1000)		Resumable statement, selected from the V$SQL view
ERROR_NUMBER	NUMBER		Error code of the last correctable error. When STATUS is set to RUNNING, its value will be set to 0.
ERROR_PARAMETER1	VARCHAR2(80)		First parameter for the error message (null if no error)
ERROR_PARAMETER2	VARCHAR2(80)		Second parameter for the error message (null if no error)
ERROR_PARAMETER3	VARCHAR2(80)		Third parameter for the error message (null if no error)
ERROR_PARAMETER4	VARCHAR2(80)		Forth parameter for the error message (null if no error)
ERROR_PARAMETER5	VARCHAR2(80)		Fifth parameter for the error message (null if no error)
ERROR_MSG	VARCHAR2(4000)		Error message corresponding to ERROR_NUMBER. It will be null when ERROR_NUMBER is null.

See Also: "USER_RESUMABLE" on page 5-124

DBA_REWRITE_EQUIVALENCES

DBA_REWRITE_EQUIVALENCES describes all rewrite equivalences in the database. Its columns are the same as those in ALL_REWRITE_EQUIVALENCES.

See Also: "ALL_REWRITE_EQUIVALENCES" on page 3-19

DBA_RGROUP

DBA_RGROUP lists all refresh groups.

Column	Datatype	NULL	Description
REFGROUP	NUMBER		Internal identifier of refresh group
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object in the refresh group
NAME	VARCHAR2(30)	NOT NULL	Name of the object in the refresh group

Column	Datatype	NULL	Description
IMPLICIT_DESTROY	VARCHAR2(1)		(Y or N) If Y, then destroy the refresh group when its last item is removed
PUSH_DEFERRED_RPC	VARCHAR2(1)		(Y or N) If Y, then push changes from snapshot to master before refresh
REFRESH_AFTER_ERRORS	VARCHAR2(1)		(Y or N) If Y, proceed with refresh despite error when pushing deferred RPC's
ROLLBACK_SEG	VARCHAR2(30)		Name of the rollback segment to use while refreshing
JOB	NUMBER	NOT NULL	Identifier of job used to refresh the group automatically
PURGE_OPTION	NUMBER(38)		The method for purging the transaction queue after each push. 1 indicates quick purge option; 2 indicates precise purge option
PARALLELISM	NUMBER(38)		The level of parallelism for transaction propagation
HEAP_SIZE	NUMBER(38)		The size of the heap

DBA_ROLE_PRIVS

DBA_ROLE_PRIVS describes the roles granted to all users and roles in the database.

Related View

USER_ROLE_PRIVS describes the roles granted to the current user.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)		Name of the user or role receiving the grant
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Granted role name
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN OPTION (YES) or not (NO)
DEFAULT_ROLE	VARCHAR2(3)		Indicates whether the role is designated as a DEFAULT ROLE for the user (YES) or not (NO)

See Also: "USER_ROLE_PRIVS" on page 5-124

DBA_ROLES

DBA_ROLES lists all roles that exist in the database.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Role name
PASSWORD_REQUIRED	VARCHAR2(8)		Indicates if the role requires a password to be enabled

DBA_ROLLBACK_SEGS

DBA_ROLLBACK_SEGS describes rollback segments.

Column	Datatype	NULL	Description
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the rollback segment
OWNER	VARCHAR2(6)		Owner of the rollback segment
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the rollback segment
SEGMENT_ID	NUMBER	NOT NULL	ID number of the rollback segment

Column	Datatype	NULL	Description
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the segment head
BLOCK_ID	NUMBER	NOT NULL	ID number of the block containing the segment header
INITIAL_EXTENT	NUMBER		Initial extent size in bytes
NEXT_EXTENT	NUMBER		Secondary extent size in bytes
MIN_EXTENTS	NUMBER	NOT NULL	Minimum number of extents
MAX_EXTENTS	NUMBER	NOT NULL	Maximum number of extent
PCT_INCREASE	NUMBER		Percent increase for extent size
STATUS	VARCHAR2(16)		Rollback segment status
INSTANCE_NUM	VARCHAR2(40)		Rollback segment owning Real Application Clusters instance number
RELATIVE_FNO	NUMBER	NOT NULL	Relative file number of the segment header

DBA_RSRC_CATEGORIES

DBA_RSRC_CATEGORIES displays all resource consumer group categories.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)		Name of the consumer group category
COMMENTS	VARCHAR2(2000)		Text comment on the consumer group category
STATUS	VARCHAR2(30)		Indicates whether the consumer group category is part of the pending area (PENDING) or not (NULL)
MANDATORY	VARCHAR2(3)		Indicates whether the consumer group category is mandatory (YES) or not (NO)

DBA_RSRC_CONSUMER_GROUP_PRIVS

DBA_RSRC_CONSUMER_GROUP_PRIVS displays information about all resource consumer groups and the users and roles assigned to them. The grant referred to in this view and the related view is the grant of the SWITCH_CONSUMER_GROUP object privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. This privilege is not granted through the GRANT SQL statement.

Related View

USER_RSRC_CONSUMER_GROUP_PRIVS displays information about the resource consumer groups to which the current user is assigned. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
GRANTED_GROUP	VARCHAR2(30)		Granted consumer group name
GRANT_OPTION	VARCHAR2(3)		Indicates whether the grant was with the GRANT option (YES) or not (NO)
INITIAL_GROUP	VARCHAR2(3)		Indicates whether the consumer group is designated as the default for this user or role (YES) or not (NO)

See Also:

- "USER_RSRC_CONSUMER_GROUP_PRIVS" on page 5-124
- *Oracle Database PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_CONSUMER_GROUPS

`DBA_RSRC_CONSUMER_GROUPS` displays information about all resource consumer groups in the database.

Column	Datatype	NULL	Description
`CONSUMER_GROUP_ID`	`NUMBER`	`NOT NULL`	ID of the consumer group
`CONSUMER_GROUP`	`VARCHAR2(30)`		Name of the consumer group
`CPU_METHOD`	`VARCHAR2(30)`		CPU resource allocation method for the consumer group
`MGMT_METHOD`	`VARCHAR2(30)`		Resource allocation method for the consumer group
`INTERNAL_USE`	`VARCHAR2(3)`		Indicates whether the consumer group is for internal use only (`YES`) or not (`NO`)
`COMMENTS`	`VARCHAR2(2000)`		Text comment on the consumer group
`CATEGORY`	`VARCHAR2(30)`		Category of the consumer group
`STATUS`	`VARCHAR2(30)`		Indicates whether the consumer group is part of the pending area (`PENDING`) or not (NULL)
`MANDATORY`	`VARCHAR2(3)`		Indicates whether the consumer group is mandatory (`YES`) or not (`NO`)

DBA_RSRC_GROUP_MAPPINGS

`DBA_RSRC_GROUP_MAPPINGS` displays the mapping between session attributes and consumer groups in the database.

Column	Datatype	NULL	Description
`ATTRIBUTE`	`VARCHAR2(30)`		Session attribute to match
`VALUE`	`VARCHAR2(128)`		Attribute value
`CONSUMER_GROUP`	`VARCHAR2(30)`		Target consumer group name
`STATUS`	`VARCHAR2(30)`		Indicates whether the consumer group is part of the pending area (`PENDING`) or not (NULL)

DBA_RSRC_IO_CALIBRATE

`DBA_RSRC_IO_CALIBRATE` displays I/O calibration results for the latest calibration run.

Column	Datatype	NULL	Description
`START_TIME`	`TIMESTAMP(6)`		Start time of the most recent I/O calibration
`END_TIME`	`TIMESTAMP(6)`		End time of the most recent I/O calibration
`MAX_IOPS`	`NUMBER`		Maximum number of data block read requests that can be sustained per second
`MAX_MBPS`	`NUMBER`		Maximum megabytes per second of maximum-sized read requests that can be sustained
`MAX_PMBPS`	`NUMBER`		Maximum megabytes per second of large I/O requests that can be sustained by a single process

Column	Datatype	NULL	Description
LATENCY	NUMBER		Latency for data block read requests
NUM_PHYSICAL_DISKS	NUMBER		Number of physical disks in the storage subsystem (as specified by the user)

DBA_RSRC_MANAGER_SYSTEM_PRIVS

DBA_RSRC_MANAGER_SYSTEM_PRIVS displays information about all the users and roles that have been granted the ADMINISTER_RESOURCE_MANAGER system privilege, which is granted using the DBMS_RESOURCE_MANAGER_PRIVS package. This privilege is not granted through the GRANT SQL statement.

Related View

USER_RSRC_MANAGER_SYSTEM_PRIVS displays information about the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. This view does not display the GRANTEE column.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	User or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the system privilege
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN option (YES) or not (NO)

See Also:

- "USER_RSRC_MANAGER_SYSTEM_PRIVS" on page 5-124
- *Oracle Database PL/SQL Packages and Types Reference* for more information on consumer group privileges

DBA_RSRC_MAPPING_PRIORITY

DBA_RSRC_MAPPING_PRIORITY displays information about all consumer group mapping attribute priorities.

Column	Datatype	NULL	Description
ATTRIBUTE	VARCHAR2(30)		Session attribute
PRIORITY	NUMBER		Priority (1 is the highest)
STATUS	VARCHAR2(30)		Indicates whether the consumer group is part of the pending area (PENDING) or not (NULL)

DBA_RSRC_PLAN_DIRECTIVES

DBA_RSRC_PLAN_DIRECTIVES displays information about all resource plan directives in the database.

Column	Datatype	NULL	Description
PLAN	VARCHAR2(30)		Name of the plan to which the directive belongs
GROUP_OR_SUBPLAN	VARCHAR2(30)		Name of the consumer group or subplan referred to
TYPE	VARCHAR2(14)		Indicates whether GROUP_OR_SUBPLAN refers to a consumer group (CONSUMER_GROUP) or a plan (PLAN)

Column	Datatype	NULL	Description
CPU_P1	NUMBER		First parameter for the CPU resource allocation method
CPU_P2	NUMBER		Second parameter for the CPU resource allocation method
CPU_P3	NUMBER		Third parameter for the CPU resource allocation method
CPU_P4	NUMBER		Fourth parameter for the CPU resource allocation method
CPU_P5	NUMBER		Fifth parameter for the CPU resource allocation method
CPU_P6	NUMBER		Sixth parameter for the CPU resource allocation method
CPU_P7	NUMBER		Seventh parameter for the CPU resource allocation method
CPU_P8	NUMBER		Eighth parameter for the CPU resource allocation method
MGMT_P1	NUMBER		First parameter for the resource allocation method
MGMT_P2	NUMBER		Second parameter for the resource allocation method
MGMT_P3	NUMBER		Third parameter for the resource allocation method
MGMT_P4	NUMBER		Fourth parameter for the resource allocation method
MGMT_P5	NUMBER		Fifth parameter for the resource allocation method
MGMT_P6	NUMBER		Sixth parameter for the resource allocation method
MGMT_P7	NUMBER		Seventh parameter for the resource allocation method
MGMT_P8	NUMBER		Eighth parameter for the resource allocation method
ACTIVE_SESS_POOL_P1	NUMBER		First parameter for the active session pool resource allocation method
QUEUING_P1	NUMBER		First parameter for the queuing resource allocation method
PARALLEL_DEGREE_LIMIT_P1	NUMBER		First parameter for the parallel degree limit resource allocation method
SWITCH_GROUP	VARCHAR2(30)		Group to switch to once the switch time is reached
SWITCH_FOR_CALL	VARCHAR2(5)		Indicates whether to switch back to the initial consumer group once the top call has completed (TRUE) or not (FALSE)
SWITCH_TIME	NUMBER		Amount of run time before the session is automatically switched
SWITCH_IO_MEGABYTES	NUMBER		Maximum megabytes of I/O within a group
SWITCH_IO_REQS	NUMBER		Maximum I/O requests within a group
SWITCH_ESTIMATE	VARCHAR2(5)		Indicates whether estimated execution time should be used for switch criteria (TRUE) or not (FALSE)
MAX_EST_EXEC_TIME	NUMBER		Maximum estimated execution time
UNDO_POOL	NUMBER		Undo pool size for the consumer group
MAX_IDLE_TIME	NUMBER		Maximum idle time for the session
MAX_IDLE_BLOCKER_TIME	NUMBER		Maximum idle time for the session when blocking other sessions
SWITCH_TIME_IN_CALL	NUMBER		Call switch time limit for execution in a group
COMMENTS	VARCHAR2(2000)		Text comment on the plan directive
STATUS	VARCHAR2(30)		Indicates whether the plan directive is part of the pending area (PENDING) or not (NULL)
MANDATORY	VARCHAR2(3)		Indicates whether the plan directive is mandatory (YES) or not (NO)

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans in general
- *Oracle Database PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

DBA_RSRC_PLANS

DBA_RSRC_PLANS displays information about all resource plans in the database. For a list of currently active plans, see "V$RSRC_PLAN" on page 7-88.

Column	Datatype	NULL	Description
PLAN_ID	NUMBER	NOT NULL	ID number of the resource plan
PLAN	VARCHAR2(30)		Name of the resource plan
NUM_PLAN_DIRECTIVES	NUMBER		Number of plan directives for the plan
CPU_METHOD	VARCHAR2(30)		CPU resource allocation method for the plan
MGMT_METHOD	VARCHAR2(30)		Resource allocation method for the plan
ACTIVE_SESS_POOL_MTH	VARCHAR2(30)		Active session pool resource allocation method for the plan
PARALLEL_DEGREE_LIMIT_MTH	VARCHAR2(30)		Parallel degree limit resource allocation method for the plan
QUEUING_MTH	VARCHAR2(30)		Queuing resource allocation method for the plan
SUB_PLAN	VARCHAR2(3)		Indicates whether the plan is a subplan (YES) or not (NO)
COMMENTS	VARCHAR2(2000)		Text comment on the plan
STATUS	VARCHAR2(30)		Indicates whether the plan is part of the pending area (PENDING) or not (NULL)
MANDATORY	VARCHAR2(3)		Indicates whether the plan is mandatory (YES) or not (NO)

See Also:

- *Oracle Database Administrator's Guide* for information on resource plans in general
- *Oracle Database PL/SQL Packages and Types Reference* for more information on creating resource plans with the DBMS_RESOURCE_MANAGER package

DBA_RULE_SET_RULES

DBA_RULE_SET_RULES describes the rules in all rule sets in the database. Its columns are the same as those in ALL_RULE_SET_RULES.

See Also: "ALL_RULE_SET_RULES" on page 3-20

DBA_RULE_SETS

DBA_RULE_SETS describes all rule sets in the database. Its columns are the same as those in ALL_RULE_SETS.

See Also: "ALL_RULE_SETS" on page 3-21

DBA_RULES

DBA_RULES describes all rules in the database. Its columns are the same as those in ALL_RULES.

See Also: "ALL_RULES" on page 3-21

DBA_SCHEDULER_CHAIN_RULES

DBA_SCHEDULER_CHAIN_RULES displays information about the rules for all chains in the database. Its columns are the same as those in ALL_SCHEDULER_CHAIN_RULES.

See Also: "ALL_SCHEDULER_CHAIN_RULES" on page 3-22

DBA_SCHEDULER_CHAIN_STEPS

DBA_SCHEDULER_CHAIN_STEPS displays information about the defined steps of all chains in the database. Its columns are the same as those in ALL_SCHEDULER_CHAIN_STEPS.

See Also: "ALL_SCHEDULER_CHAIN_STEPS" on page 3-22

DBA_SCHEDULER_CHAINS

DBA_SCHEDULER_CHAINS displays information about all chains in the database. Its columns are the same as those in ALL_SCHEDULER_CHAINS.

See Also: "ALL_SCHEDULER_CHAINS" on page 3-23

DBA_SCHEDULER_CREDENTIALS

DBA_SCHEDULER_CREDENTIALS displays information about all credentials in the database. Its columns are the same as those in ALL_SCHEDULER_CREDENTIALS.

See Also: "ALL_SCHEDULER_CREDENTIALS" on page 3-24

DBA_SCHEDULER_GLOBAL_ATTRIBUTE

DBA_SCHEDULER_GLOBAL_ATTRIBUTE displays the values of all scheduler attributes in the database (for example, DEFAULT_TIMEZONE and CURRENT_OPEN_WINDOW). Its columns are the same as those in ALL_SCHEDULER_GLOBAL_ATTRIBUTE.

See Also: "ALL_SCHEDULER_GLOBAL_ATTRIBUTE" on page 3-24

DBA_SCHEDULER_JOB_ARGS

DBA_SCHEDULER_JOB_ARGS displays information about the arguments of all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_ARGS.

See Also: "ALL_SCHEDULER_JOB_ARGS" on page 3-25

DBA_SCHEDULER_JOB_CLASSES

DBA_SCHEDULER_JOB_CLASSES displays information about all Scheduler job classes in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_CLASSES.

See Also: "ALL_SCHEDULER_JOB_CLASSES" on page 3-25

DBA_SCHEDULER_JOB_LOG

DBA_SCHEDULER_JOB_LOG displays log information for all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_LOG.

See Also: "ALL_SCHEDULER_JOB_LOG" on page 3-26

DBA_SCHEDULER_JOB_ROLES

DBA_SCHEDULER_JOB_ROLES displays information about all Scheduler jobs in the database by database role.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	Owner of the Scheduler job
JOB_NAME	VARCHAR2(30)	NOT NULL	Name of the Scheduler job
JOB_SUBNAME	VARCHAR2(30)		Subname of the Scheduler job (for a job running a chain step)
JOB_CREATOR	VARCHAR2(30)		Creator of the Scheduler job
DATABASE_ROLE	VARCHAR2(16)		Name of the database role
PROGRAM_OWNER	VARCHAR2(4000)		Owner of the program associated with the job
PROGRAM_NAME	VARCHAR2(4000)		Name of the program associated with the job
JOB_TYPE	VARCHAR2(16)		Inline job action type: ■ PLSQL_BLOCK ■ STORED_PROCEDURE ■ EXECUTABLE ■ CHAIN
JOB_ACTION	VARCHAR2(4000)		Inline job action
JOB_CLASS	VARCHAR2(30)		Name of the job class associated with the job
SCHEDULE_OWNER	VARCHAR2(4000)		Owner of the schedule that the job uses (can be a window or a window group)
SCHEDULE_NAME	VARCHAR2(4000)		Name of the schedule that the job uses (can be a window or a window group)
SCHEDULE_TYPE	VARCHAR2(12)		Type of the schedule that the job uses: ■ IMMEDIATE - Start date and repeat interval are NULL ■ ONCE - Repeat interval is NULL ■ PLSQL - PL/SQL expression used as schedule ■ CALENDAR - Oracle calendaring expression used as schedule ■ EVENT - Event schedule ■ NAMED - Named schedule ■ WINDOW - Window used as schedule ■ WINDOW_GROUP - Window group used as schedule

Column	Datatype	NULL	Description
START_DATE	TIMESTAMP(6) WITH TIME ZONE		Original scheduled start date of the job (for an inline schedule)
REPEAT_INTERVAL	VARCHAR2(4000)		Inline schedule PL/SQL expression or calendar string
END_DATE	TIMESTAMP(6) WITH TIME ZONE		Date after which the job will no longer run (for an inline schedule)
LAST_START_DATE	TIMESTAMP(6) WITH TIME ZONE		Last date on which the job ran
ENABLED	VARCHAR2(5)		Indicates whether the job is enabled (TRUE) or disabled (FALSE)
STATE	VARCHAR2(15)		Current state of the job: ■ DISABLED ■ RETRY SCHEDULED ■ SCHEDULED ■ RUNNING ■ COMPLETED ■ BROKEN ■ FAILED ■ REMOTE ■ SUCCEEDED ■ CHAIN_STALLED
COMMENTS	VARCHAR2(240)		Comments on the job

DBA_SCHEDULER_JOB_RUN_DETAILS

DBA_SCHEDULER_JOB_RUN_DETAILS displays log run details for all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOB_RUN_DETAILS.

See Also: "ALL_SCHEDULER_JOB_RUN_DETAILS" on page 3-27

DBA_SCHEDULER_JOBS

DBA_SCHEDULER_JOBS displays information about all Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_JOBS.

See Also: "ALL_SCHEDULER_JOBS" on page 3-28

DBA_SCHEDULER_PROGRAM_ARGS

DBA_SCHEDULER_PROGRAM_ARGS displays information about the arguments of all Scheduler programs in the database. Its columns are the same as those in ALL_SCHEDULER_PROGRAM_ARGS.

See Also: "ALL_SCHEDULER_PROGRAM_ARGS" on page 3-31

DBA_SCHEDULER_PROGRAMS

DBA_SCHEDULER_PROGRAMS displays information about all Scheduler programs in the database. Its columns are the same as those in ALL_SCHEDULER_PROGRAMS.

See Also: "ALL_SCHEDULER_PROGRAMS" on page 3-32

DBA_SCHEDULER_REMOTE_DATABASES

DBA_SCHEDULER_REMOTE_DATABASES displays information about all remote databases that have been registered as sources and destinations for remote database jobs. Its columns are the same as those in ALL_SCHEDULER_REMOTE_DATABASES.

See Also: "ALL_SCHEDULER_REMOTE_DATABASES" on page 3-33

DBA_SCHEDULER_REMOTE_JOBSTATE

DBA_SCHEDULER_REMOTE_JOBSTATE displays information about the state of all jobs at remote databases. Its columns are the same as those in ALL_SCHEDULER_REMOTE_JOBSTATE.

See Also: "ALL_SCHEDULER_REMOTE_JOBSTATE" on page 3-33

DBA_SCHEDULER_RUNNING_CHAINS

DBA_SCHEDULER_RUNNING_CHAINS displays information about the chain steps of all running chains in the database. Its columns are the same as those in ALL_SCHEDULER_RUNNING_CHAINS.

See Also: "ALL_SCHEDULER_RUNNING_CHAINS" on page 3-34

DBA_SCHEDULER_RUNNING_JOBS

DBA_SCHEDULER_RUNNING_JOBS displays information about all running Scheduler jobs in the database. Its columns are the same as those in ALL_SCHEDULER_RUNNING_JOBS.

See Also: "ALL_SCHEDULER_RUNNING_JOBS" on page 3-35

DBA_SCHEDULER_SCHEDULES

DBA_SCHEDULER_SCHEDULES displays information about all Scheduler schedules in the database. Its columns are the same as those in ALL_SCHEDULER_SCHEDULES.

See Also: "ALL_SCHEDULER_SCHEDULES" on page 3-36

DBA_SCHEDULER_WINDOW_DETAILS

DBA_SCHEDULER_WINDOW_DETAILS displays log details for all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_DETAILS.

See Also: "ALL_SCHEDULER_WINDOW_DETAILS" on page 3-37

DBA_SCHEDULER_WINDOW_GROUPS

DBA_SCHEDULER_WINDOW_GROUPS displays information about all Scheduler window groups in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_GROUPS.

See Also: "ALL_SCHEDULER_WINDOW_GROUPS" on page 3-37

DBA_SCHEDULER_WINDOW_LOG

DBA_SCHEDULER_WINDOW_LOG displays log information for all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOW_LOG.

See Also: "ALL_SCHEDULER_WINDOW_LOG" on page 3-38

DBA_SCHEDULER_WINDOWS

DBA_SCHEDULER_WINDOWS displays information about all Scheduler windows in the database. Its columns are the same as those in ALL_SCHEDULER_WINDOWS.

See Also: "ALL_SCHEDULER_WINDOWS" on page 3-38

DBA_SCHEDULER_WINGROUP_MEMBERS

DBA_SCHEDULER_WINGROUP_MEMBERS displays the members of all Scheduler window groups in the database. Its columns are the same as those in ALL_SCHEDULER_WINGROUP_MEMBERS.

See Also: "ALL_SCHEDULER_WINGROUP_MEMBERS" on page 3-39

DBA_SEC_RELEVANT_COLS

DBA_SEC_RELEVANT_COLS describes the security relevant columns of all security policies in the database. Its columns are the same as those in ALL_SEC_RELEVANT_COLS.

See Also: "ALL_SEC_RELEVANT_COLS" on page 3-40

DBA_SECONDARY_OBJECTS

DBA_SECONDARY_OBJECTS provides information about all secondary objects that are associated with domain indexes in the database. This view is only relevant in the context of domain indexes. Its columns are the same as those in "ALL_SECONDARY_OBJECTS" on page 3-40.

DBA_SEGMENTS

DBA_SEGMENTS describes the storage allocated for all segments in the database.

Related View

USER_SEGMENTS describes the storage allocated for the segments owned by the current user's objects. This view does not display the OWNER, HEADER_FILE, HEADER_BLOCK, or RELATIVE_FNO columns.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the segment owner
SEGMENT_NAME	VARCHAR2(81)		Name, if any, of the segment
PARTITION_NAME	VARCHAR2(30)		Object Partition Name (Set to NULL for non-partitioned objects)

Column	Datatype	NULL	Description
SEGMENT_TYPE	VARCHAR2(18)		Type of segment: - NESTED TABLE - TABLE - TABLE PARTITION - CLUSTER - LOBINDEX - INDEX - INDEX PARTITION - LOBSEGMENT - TABLE SUBPARTITION - INDEX SUBPARTITION - LOB PARTITION - LOB SUBPARTITION - ROLLBACK - TYPE2 UNDO - DEFERRED ROLLBACK - TEMPORARY - CACHE - SPACE HEADER - UNDEFINED
SEGMENT_SUBTYPE	VARCHAR2(10)		Subtype of LOB segment: SECUREFILE, ASSM, MSSM, and NULL
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
BYTES	NUMBER		Size, in bytes, of the segment
BLOCKS	NUMBER		Size, in Oracle blocks, of the segment
EXTENTS	NUMBER		Number of extents allocated to the segment
INITIAL_EXTENT	NUMBER		Size in bytes requested for the initial extent of the segment at create time. (Oracle rounds the extent size to multiples of 5 blocks if the requested size is greater than 5 blocks.)
NEXT_EXTENT	NUMBER		Size in bytes of the next extent to be allocated to the segment
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
MAX_SIZE	NUMBER		Maximum number of blocks allowed in the segment
RETENTION	VARCHAR2(7)		Retention option for SECUREFILE segment
MINRETENTION	NUMBER		Minimum retention duration for SECUREFILE segment
PCT_INCREASE	NUMBER		Percent by which to increase the size of the next extent to be allocated
FREELISTS	NUMBER		Number of process freelists allocated to this segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
RELATIVE_FNO	NUMBER		Relative file number of the segment header
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the object

See Also: "USER_SEGMENTS" on page 5-127

DBA_SEGMENTS_OLD

DBA_SEGMENTS_OLD lists information about storage allocated for all database segments.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Username of the segment owner
SEGMENT_NAME	VARCHAR2(81)		Name, if any, of the segment
PARTITION_NAME	VARCHAR2(30)		Name of the partition
SEGMENT_TYPE	VARCHAR2(18)		Type of segment: INDEX PARTITION, TABLE PARTITION, TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE, LOBSEGMENT and LOBINDEX
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the segment
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
BYTES	NUMBER		Size, in bytes, of the segment
BLOCKS	NUMBER		Size, in Oracle blocks, of the segment
EXTENTS	NUMBER		Number of extents allocated to the segment
INITIAL_EXTENT	NUMBER		Size in bytes requested for the initial extent of the segment at create time. (Oracle rounds the extent size to multiples of 5 blocks if the requested size is greater than 5 blocks.)
NEXT_EXTENT	NUMBER		Size in bytes of the next extent to be allocated to the segment
MIN_EXTENTS	NUMBER		Minimum number of extents allowed in the segment
MAX_EXTENTS	NUMBER		Maximum number of extents allowed in the segment
PCT_INCREASE	NUMBER		Percent by which to increase the size of the next extent to be allocated
FREELISTS	NUMBER		Number of process freelists allocated to the segment
FREELIST_GROUPS	NUMBER		Number of freelist groups allocated to this segment
RELATIVE_FNO	NUMBER		Relative file number of the segment header
BUFFER_POOL	VARCHAR2(7)		Default buffer pool for the object

DBA_SEQUENCES

DBA_SEQUENCES describes all sequences in the database. Its columns are the same as those in "ALL_SEQUENCES" on page 3-41.

DBA_SERVER_REGISTRY

DBA_SERVER_REGISTRY displays information about the components loaded into the database.

Column	Datatype	NULL	Description
COMP_ID	VARCHAR2(30)	NOT NULL	Component identifier
COMP_NAME	VARCHAR2(255)		Component name
VERSION	VARCHAR2(30)		Component version loaded

Column	Datatype	NULL	Description
STATUS	VARCHAR2(11)		Component status: ■ INVALID ■ VALID ■ LOADING ■ LOADED ■ UPGRADING ■ UPGRADED ■ DOWNGRADING ■ DOWNGRADED ■ REMOVING ■ REMOVED
MODIFIED	VARCHAR2(20)		Time when the component was last modified
CONTROL	VARCHAR2(30)	NOT NULL	User that created the component entry
SCHEMA	VARCHAR2(30)	NOT NULL	User that contains the objects for the component
PROCEDURE	VARCHAR2(61)		Validation procedure
STARTUP	VARCHAR2(8)		Indicates whether the component requires a startup after the upgrade (REQUIRED) or not
PARENT_ID	VARCHAR2(30)		Parent component identifier

DBA_SERVICES

DBA_SERVICES displays all services in the database. The view excludes rows marked for deletion. Its columns are the same as those in ALL_SERVICES.

DBA_SOURCE

DBA_SOURCE describes the text source of all stored objects in the database. Its columns are the same as those in "ALL_SOURCE" on page 3-42.

DBA_SOURCE_TABLES

DBA_SOURCE_TABLES describes all source tables in the database for which a change table is defined. This view is intended for use by Change Data Capture publishers.

Related View

USER_SOURCE_TABLES describes the source tables owned by the current user for which a change table is defined.

Column	Datatype	NULL	Description
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Table owner in the source database
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Table name in the source database

See Also: "USER_SOURCE_TABLES" on page 5-128

DBA_SQL_MANAGEMENT_CONFIG

DBA_SQL_MANAGEMENT_CONFIG displays the configuration parameters of the SQL management base. You must have the DBA role in order to change the configuration parameter values.

Column	Datatype	NULL	Description
PARAMETER_NAME	VARCHAR2(30)	NOT NULL	Name of the configuration parameter: ▪ SPACE_BUDGET_PERCENT ▪ PLAN_RETENTION_WEEKS
PARAMETER_VALUE	NUMBER	NOT NULL	Value of the configuration parameter
LAST_MODIFIED	TIMESTAMP(6)		Time the parameter value was last updated
MODIFIED_BY	VARCHAR2(30)		User who last updated the parameter value

DBA_SQL_PATCHES

DBA_SQL_PATCHES displays the set of SQL patches.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the SQL patch
CATEGORY	VARCHAR2(30)	NOT NULL	Category of the SQL patch
SIGNATURE	NUMBER	NOT NULL	Unique identifier generated from normalized SQL text
SQL_TEXT	CLOB	NOT NULL	Un-normalized SQL text
CREATED	TIMESTAMP(6)	NOT NULL	Timestamp when the SQL patch was created
LAST_MODIFIED	TIMESTAMP(6)		Timestamp when the SQL patch was last modified
DESCRIPTION	VARCHAR2(500)		Text description provided for the SQL patch
STATUS	VARCHAR2(8)		Status of the SQL patch: ▪ ENABLED ▪ DISABLED
FORCE_MATCHING	VARCHAR2(3)		Indicates whether the signature is force matching (YES) or exact matching (NO)
TASK_ID	NUMBER		Advisor task ID that generated the SQL patch
TASK_EXEC_NAME	VARCHAR2(30)		Advisor execution name for the SQL patch
TASK_OBJ_ID	NUMBER		Advisor object ID for the SQL patch
TASK_FND_ID	NUMBER		Advisor finding ID for the SQL patch
TASK_REC_ID	NUMBER		Advisor recommendation ID for the SQL patch

DBA_SQL_PLAN_BASELINES

DBA_SQL_PLAN_BASELINES displays information about the SQL plan baselines currently created for specific SQL statements.

Column	Datatype	NULL	Description
SIGNATURE	NUMBER	NOT NULL	Unique SQL identifier generated from normalized SQL text
SQL_HANDLE	VARCHAR2(30)	NOT NULL	Unique SQL identifier in string form as a search key
SQL_TEXT	CLOB	NOT NULL	Un-normalized SQL text
PLAN_NAME	VARCHAR2(30)	NOT NULL	Unique plan identifier in string form as a search key
CREATOR	VARCHAR2(30)		User who created the plan baseline

Column	Datatype	NULL	Description
ORIGIN	VARCHAR2(14)		How the plan baseline was created: ■ MANUAL-LOAD ■ AUTO-CAPTURE ■ MANUAL-SQLTUNE ■ AUTO-SQLTUNE
PARSING_SCHEMA_NAME	VARCHAR2(30)		Name of the parsing schema
DESCRIPTION	VARCHAR2(500)		Text description provided for the plan baseline
VERSION	VARCHAR2(64)		Database version at the time of plan baseline creation
CREATED	TIMESTAMP(6)	NOT NULL	Timestamp when the plan baseline was created
LAST_MODIFIED	TIMESTAMP(6)		Timestamp when the plan baseline was last modified
LAST_EXECUTED	TIMESTAMP(6)		Timestamp when the plan baseline was last executed
LAST_VERIFIED	TIMESTAMP(6)		Timestamp when the plan baseline was last verified
ENABLED	VARCHAR2(3)		Indicates whether the plan baseline is enabled (YES) or disabled (NO)
ACCEPTED	VARCHAR2(3)		Indicates whether the plan baseline is accepted (YES) or not (NO)
FIXED	VARCHAR2(3)		Indicates whether the plan baseline is fixed (YES) or not (NO)
AUTOPURGE	VARCHAR2(3)		Indicates whether the plan baseline is auto-purged (YES) or not (NO)
OPTIMIZER_COST	NUMBER		Optimizer cost at the time the plan baseline was created
MODULE	VARCHAR2(48)		Application module name
ACTION	VARCHAR2(32)		Application action
EXECUTIONS	NUMBER		Number of executions at the time the plan baseline was created
ELAPSED_TIME	NUMBER		Total elapsed time at the time the plan baseline was created
CPU_TIME	NUMBER		Total CPU time at the time the plan baseline was created
BUFFER_GETS	NUMBER		Total buffer gets at the time the plan baseline was created
DISK_READS	NUMBER		Total disk reads at the time the plan baseline was created
DIRECT_WRITES	NUMBER		Total direct writes at the time the plan baseline was created
ROWS_PROCESSED	NUMBER		Total rows processed at the time the plan baseline was created
FETCHES	NUMBER		Total number of fetches at the time the plan baseline was created
END_OF_FETCH_COUNT	NUMBER		Total number of full fetches at the time the plan baseline was created

See Also:

- The SQL Plan Management chapter in *Oracle Database Performance Tuning Guide*
- The DBMS_SQLTUNE package in *Oracle Database PL/SQL Packages and Types Reference*

DBA_SQL_PROFILES

DBA_SQL_PROFILES displays information about SQL profiles currently created for specific SQL statements.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the SQL profile
CATEGORY	VARCHAR2(30)	NOT NULL	Category of the SQL profile
SIGNATURE	NUMBER	NOT NULL	Unique identifier generated from normalized SQL text
SQL_TEXT	CLOB	NOT NULL	Un-normalized SQL text
CREATED	DATE	NOT NULL	Timestamp when the SQL profile was created
LAST_MODIFIED	DATE	NOT NULL	Timestamp when the SQL profile was last modified
DESCRIPTION	VARCHAR2(500)		Text description provided for the SQL profile
TYPE	VARCHAR2(9)		Type of the SQL profile (how it was created): ■ MANUAL ■ AUTO-TUNE
STATUS	VARCHAR2(8)		Status of the SQL profile: ■ ENABLED ■ DISABLED ■ VOID
FORCE_MATCHING	VARCHAR2(3)		If TRUE this causes SQL Profiles to target all SQL statements which have the same text after normalizing all literal values to bind variables. (Note that if a combination of literal values and bind variables is used in the same SQL text, no transformation occurs). This is analogous to the matching algorithm use by the FORCE option of the CURSOR_SHARING parameter. If FALSE, literals are not transformed. This is analogous to the matching algorithm used by the EXACT option of the CURSOR_SHARING parameter.
TASK_ID	NUMBER		Advisor task ID that generated the SQL profile
EXECUTION_NAME	VARCHAR2(30)		Advisor execution name for the SQL profile
OBJECT_ID	NUMBER		Advisor object ID for the SQL profile
FINDING_ID	NUMBER		Advisor finding ID for the SQL profile
REC_ID	NUMBER		Advisor recommendation ID for the SQL profile

See Also: The DBMS_SQLTUNE package in *Oracle Database PL/SQL Packages and Types Reference*

DBA_SQLJ_TYPE_ATTRS

DBA_SQLJ_TYPE_ATTRS describes the attributes of all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPE_ATTRS.

See Also: "ALL_SQLJ_TYPE_ATTRS" on page 3-42

DBA_SQLJ_TYPE_METHODS

DBA_SQLJ_TYPE_METHODS describes the methods of all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPE_METHODS.

See Also: "ALL_SQLJ_TYPE_METHODS" on page 3-43

DBA_SQLJ_TYPES

DBA_SQLJ_TYPES describes all SQLJ object types in the database. Its columns are the same as those in ALL_SQLJ_TYPES.

See Also: "ALL_SQLJ_TYPES" on page 3-44

DBA_SQLSET

DBA_SQLSET displays information about all SQL tuning sets in the database. Its columns are the same as those in "ALL_SQLSET" on page 3-45.

DBA_SQLSET_BINDS

DBA_SQLSET_BINDS displays the bind values associated with all SQL tuning sets in the database. Its columns are the same as those in "ALL_SQLSET_BINDS" on page 3-45.

DBA_SQLSET_PLANS

DBA_SQLSET_PLANS describes captured plans in the SQL tuning sets in the database. Its columns are the same as those in "ALL_SQLSET_PLANS" on page 3-46.

DBA_SQLSET_REFERENCES

DBA_SQLSET_REFERENCES describes whether or not all SQL tuning sets in the database are active. A SQL tuning set cannot be dropped if it is referenced. Its columns are the same as those in "ALL_SQLSET_REFERENCES" on page 3-49.

DBA_SQLSET_STATEMENTS

DBA_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form all SQL tuning sets in the database. Its columns are the same as those in "ALL_SQLSET_STATEMENTS" on page 3-49.

DBA_SQLTUNE_BINDS

DBA_SQLTUNE_BINDS displays the bind values associated with all tuned SQL statements in the database.

Related View

USER_SQLTUNE_BINDS displays the bind values associated with the tuned SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
POSITION	NUMBER(38)	NOT NULL	Bind position
VALUE	ANYDATA		Bind value

See Also: "USER_SQLTUNE_BINDS" on page 5-129

DBA_SQLTUNE_PLANS

DBA_SQLTUNE_PLANS displays information about the execution plans generated for all SQL statements in the database during a SQL tuning session.

Related View

USER_SQLTUNE_PLANS displays information about the execution plans generated for the SQL statements owned by the current user during a SQL tuning session.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	???
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
ATTRIBUTE	VARCHAR2(27)		Type of the execution plan: ■ Original - Original plan of the query ■ Original with adjusted cost - Same as Original but with adjusted cost ■ Using SQL profile - Plan with sql profile applied ■ Using new indices - Plan with indexes applied
STATEMENT_ID	VARCHAR2(30)		Optional statement identifier specified in the EXPLAIN PLAN statement
PLAN_HASH_VALUE	NUMBER	NOT NULL	Numerical representation of the execution plan
PLAN_ID	NUMBER		Plan identifier
TIMESTAMP	DATE		Date and time when the EXPLAIN PLAN statement was issued
REMARKS	VARCHAR2(4000)		Place for comments that can be added to the steps of the execution plan
OPERATION	VARCHAR2(30)		Name of the operation performed at this step
OPTIONS	VARCHAR2(255)		Options used for the operation performed at this step
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object
OBJECT_OWNER	VARCHAR2(30)		Owner of the object
OBJECT_NAME	VARCHAR2(30)		Name of the object
OBJECT_ALIAS	VARCHAR2(65)		Object alias
OBJECT_INSTANCE	NUMBER(38)		Numbered position of the object name in the original SQL statement
OBJECT_TYPE	VARCHAR2(30)		Descriptive modifier that further describes the type of object
OPTIMIZER	VARCHAR2(255)		Current mode of the optimizer
SEARCH_COLUMNS	NUMBER		Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
ID	NUMBER(38)	NOT NULL	Identification number for this step in the execution plan
PARENT_ID	NUMBER(38)		ID of the next step that operates on the results of this step
DEPTH	NUMBER(38)		Depth
POSITION	NUMBER(38)		Order of processing for steps with the same parent ID
COST	NUMBER(38)		Cost of the current operation estimated by the cost-based optimizer (CBO)
CARDINALITY	NUMBER(38)		Number of rows returned by the current operation (estimated by the CBO)
BYTES	NUMBER(38)		Number of bytes returned by the current operation

Column	Datatype	NULL	Description
OTHER_TAG	VARCHAR2(255)		Describes the function of the SQL text in the OTHER column. Values for OTHER_TAG are: ■ SERIAL - SQL is the text of a locally-executed, serial query plan. Currently, SQL is not loaded in OTHER for this case. ■ SERIAL_FROM_REMOTE - SQL text shown in the OTHER column will be executed at a remote site ■ PARALLEL_COMBINED_WITH_PARENT - Parent of this operation is a DFO that performs both operations in the parallel execution plan ■ PARALLEL_COMBINED_WITH_CHILD - Child of this operation is a DFO that performs both operations in the parallel execution plan. ■ PARALLEL_TO_SERIAL - SQL text shown in the OTHER column is the top-level of the parallel plan. ■ PARALLEL_TO_PARALLEL - SQL text shown in the OTHER column is executed and output in parallel ■ PARALLEL_FROM_SERIAL - Operation consumes data from a serial operation and outputs it in parallel
PARTITION_START	VARCHAR2(255)		Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(255)		Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER(38)		Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Information about parallel execution servers and parallel queries
DISTRIBUTION	VARCHAR2(30)		Distribution method
CPU_COST	NUMBER(38)		User-defined CPU cost
IO_COST	NUMBER(38)		User-defined I/O cost
TEMP_SPACE	NUMBER(38)		Temporary space usage of the operation (sort or hash-join) as estimated by the CBO
ACCESS_PREDICATES	VARCHAR2(4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)		Expressions produced by the operation
TIME	NUMBER(38)		Elapsed time (in seconds) of the operation as estimated by the CBO
QBLOCK_NAME	VARCHAR2(30)		Name of the query block

See Also: "USER_SQLTUNE_PLANS" on page 5-129

DBA_SQLTUNE_RATIONALE_PLAN

DBA_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of all SQL statements in the database.

Related View

USER_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of the SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
EXECUTION_NAME	VARCHAR2(30)	NOT NULL	???
RATIONALE_ID	NUMBER(38)	NOT NULL	Rationale identifier
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
OPERATION_ID	NUMBER(38)	NOT NULL	Operation identifier
PLAN_ATTRIBUTE	VARCHAR2(27)		Type of the execution plan: ■ Original - Original plan of the query ■ Original with adjusted cost - Same as Original but with adjusted cost ■ Using SQL profile - Plan with sql profile applied ■ Using new indices - Plan with indexes applied

See Also: "USER_SQLTUNE_RATIONALE_PLAN" on page 5-129

DBA_SQLTUNE_STATISTICS

DBA_SQLTUNE_STATISTICS displays statistics associated with all SQL statements in the database.

Related View

USER_SQLTUNE_STATISTICS displays statistics associated with the SQL statements owned by the current user.

Column	Datatype	NULL	Description
TASK_ID	NUMBER(38)	NOT NULL	Tuning task identifier
OBJECT_ID	NUMBER(38)	NOT NULL	Advisor framework object identifier
PARSING_SCHEMA_ID	NUMBER		Schema under which the SQL is parsed
MODULE	VARCHAR2(48)		Last application module recorded for the SQL
ACTION	VARCHAR2(32)		Last application action recorded for the SQL
ELAPSED_TIME	NUMBER		Elapsed time for the SQL statement
CPU_TIME	NUMBER		CPU time for the SQL
BUFFER_GETS	NUMBER		Number of buffer gets
DISK_READS	NUMBER		Number of disk reads
DIRECT_WRITES	NUMBER		Number of disk writes
ROWS_PROCESSED	NUMBER		Number of rows processed by the SQL
FETCHES	NUMBER		Number of fetches
EXECUTIONS	NUMBER		Number of executions
END_OF_FETCH_COUNT	NUMBER		End of fetch count
OPTIMIZER_COST	NUMBER		Optimizer cost for the SQL
OPTIMIZER_ENV	RAW(1000)		Optimizer environment
COMMAND_TYPE	NUMBER		Command type

See Also: "USER_SQLTUNE_STATISTICS" on page 5-129

DBA_SSCR_CAPTURE

DBA_SSCR_CAPTURE displays session state capture statistics.

Column	Datatype	NULL	Description
DB_NAME	VARCHAR2(4000)		Database name of captured session
INST_NAME	VARCHAR2(4000)		Instance name of captured session
INST_ID	VARCHAR2(4000)		Instance ID of captured session
SESSION_ID	NUMBER		Session ID of captured session
SESSION_SERIAL#	NUMBER		Session serial number of captured session
USER_NAME	VARCHAR2(30)	NOT NULL	User name of captured session
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name of captured session
SEQUENCE#	NUMBER		Sequence number of captured session
CAPTURE_MODE	VARCHAR2(7)		Mode of capture operation
CAPTURE_SCOPE	VARCHAR2(7)		Scope of capture operation
CAPTURE_FORMAT	VARCHAR2(9)		Format of capture files
CAPTURE_DIR	VARCHAR2(30)		Directory object of capture files
CAPTURE_LOCATOR	RAW(64)		Locator of master capture file
CAPTURE_TIME	TIMESTAMP(6)		Timestamp of capture operation

DBA_SSCR_RESTORE

DBA_SSCR_RESTORE displays session state restore statistics.

Column	Datatype	NULL	Description
DB_NAME	VARCHAR2(4000)		Database name of restored session
INST_NAME	VARCHAR2(4000)		Instance name of restored session
INST_ID	VARCHAR2(4000)		Instance ID of restored session
SESSION_ID	NUMBER		Session ID of restored session
SESSION_SERIAL#	NUMBER		Session serial number of restored session
USER_NAME	VARCHAR2(30)	NOT NULL	User name of restored session
SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema name of restored session
SEQUENCE#	NUMBER		Sequence number of restore operation
RESTORE_MODE	VARCHAR2(7)		Mode of restore operation
RESTORE_SCOPE	VARCHAR2(7)		Scope of restore operation
RESTORE_FORMAT	VARCHAR2(9)		Format of restore files
RESTORE_DIR	VARCHAR2(30)		Directory object of restore files
RESTORE_LOCATOR	RAW(64)		Locator of master restore file
RESTORE_TIME	TIMESTAMP(6)		Timestamp of restore operation

DBA_STAT_EXTENSIONS

DBA_STAT_EXTENSIONS displays information about all optimizer statistics extensions in the database.

See Also: "ALL_STAT_EXTENSIONS" on page 3-51

DBA_STMT_AUDIT_OPTS

DBA_STMT_AUDIT_OPTS describes current system auditing options across the system and by user.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2(30)		User name if by user auditing. ANY CLIENT if access by a proxy on behalf of a client is being audited. NULL for system-wide auditing
PROXY_NAME	VARCHAR2(30)		The name of the proxy user which is performing an operation for the client. NULL if the client is performing the operation directly.
AUDIT_OPTION	VARCHAR2(40)	NOT NULL	Name of the system auditing option
SUCCESS	VARCHAR2(10)		Mode for WHENEVER SUCCESSFUL system auditing
FAILURE	VARCHAR2(10)		Mode for WHENEVER NOT SUCCESSFUL system auditing

DBA_STORED_SETTINGS

DBA_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units for which the current user has execute privileges. It also returns parameter information for all objects in the database and is accessible only to users with the SELECT_CATALOG_ROLE privilege. Its columns are the same as those in "ALL_STORED_SETTINGS" on page 3-51.

DBA_STREAMS_ADD_COLUMN

DBA_STREAMS_ADD_COLUMN displays information about new columns in the Streams capture process.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
SCHEMA_NAME	VARCHAR2(30)		Schema of the column to be added
TABLE_NAME	VARCHAR2(30)		Table of the column to be added
COLUMN_NAME	VARCHAR2(4000)		Name of the column to be added
COLUMN_VALUE	ANYDATA		Value of the column to be added
COLUMN_TYPE	VARCHAR2(4000)		Type of the column to be added
COLUMN_FUNCTION	VARCHAR2(30)		Name of the default function used to add a column
VALUE_TYPE	VARCHAR2(3)		Indicates whether to modify the old (OLD), new (NEW), or both (*) values of the LCR
PRECEDENCE	NUMBER		3 (the execution order relative to other transformations on the same STEP_NUMBER; the smaller number will be executed first)
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_ADMINISTRATOR

DBA_STREAMS_ADMINISTRATOR displays information about the users who have been granted privileges to be a Streams administrator by procedures in the DBMS_STREAMS_AUTH package.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user who has been granted privileges to be a Streams administrator
LOCAL_PRIVILEGES	VARCHAR2(3)		Indicates whether the user has been granted local Streams administrator privileges (YES) or not (NO)
ACCESS_FROM_REMOTE	VARCHAR2(3)		Indicates whether the user can be used for remote Streams administration through a database link (YES) or not (NO)

DBA_STREAMS_COLUMNS

DBA_STREAMS_COLUMNS displays information about all Streams unsupported columns in the database. Its columns are the same as those in ALL_STREAMS_COLUMNS.

See Also: "ALL_STREAMS_COLUMNS" on page 3-52

DBA_STREAMS_DELETE_COLUMN

DBA_STREAMS_DELETE_COLUMN displays information about columns to be deleted from the Streams capture process.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
SCHEMA_NAME	VARCHAR2(30)		Schema of the column to be deleted
TABLE_NAME	VARCHAR2(30)		Table of the column to be deleted
COLUMN_NAME	VARCHAR2(4000)		Name of the column to delete
VALUE_TYPE	VARCHAR2(3)		Indicates whether to modify the old (OLD), new (NEW), or both (*) values of the LCR
PRECEDENCE	NUMBER		1 (the execution order relative to other transformations on the same STEP_NUMBER; the smaller number will be executed first)
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_GLOBAL_RULES

DBA_STREAMS_GLOBAL_RULES displays information about the global rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_GLOBAL_RULES.

See Also: "ALL_STREAMS_GLOBAL_RULES" on page 3-52

DBA_STREAMS_MESSAGE_CONSUMERS

DBA_STREAMS_MESSAGE_CONSUMERS displays information about all Streams messaging clients in the database. Its columns are the same as those in ALL_STREAMS_MESSAGE_CONSUMERS.

See Also: "ALL_STREAMS_MESSAGE_CONSUMERS" on page 3-53

DBA_STREAMS_MESSAGE_RULES

DBA_STREAMS_MESSAGE_RULES displays information about all Streams messaging rules in the database. Its columns are the same as those in ALL_STREAMS_MESSAGE_RULES.

See Also: "ALL_STREAMS_MESSAGE_RULES" on page 3-54

DBA_STREAMS_NEWLY_SUPPORTED

DBA_STREAMS_NEWLY_SUPPORTED displays information about all tables in the database that are newly supported by Streams. Its columns are the same as those in ALL_STREAMS_NEWLY_SUPPORTED.

See Also: "ALL_STREAMS_NEWLY_SUPPORTED" on page 3-54

DBA_STREAMS_RENAME_COLUMN

DBA_STREAMS_RENAME_COLUMN displays information about renamed columns in the Streams capture process.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
SCHEMA_NAME	VARCHAR2(30)		Schema of the column to be renamed
TABLE_NAME	VARCHAR2(30)		Table of the column to be renamed
FROM_COLUMN_NAME	VARCHAR2(4000)		Column to rename
TO_COLUMN_NAME	VARCHAR2(4000)		New column name
VALUE_TYPE	VARCHAR2(3)		Indicates whether to modify the old (OLD), new (NEW), or both (*) values of the LCR
PRECEDENCE	NUMBER		2 (the execution order relative to other transformations on the same STEP_NUMBER; the smaller number will be executed first)
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_RENAME_SCHEMA

DBA_STREAMS_RENAME_SCHEMA displays information about renamed schemas in a Streams capture process.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
FROM_SCHEMA_NAME	VARCHAR2(30)		Schema to be renamed
TO_SCHEMA_NAME	VARCHAR2(30)		New schema name
PRECEDENCE	NUMBER		5 (the execution order relative to other transformations on the same STEP_NUMBER; the smaller number will be executed first)
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_RENAME_TABLE

DBA_STREAMS_RENAME_TABLE displays information about renamed tables in a Streams capture process.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
FROM_SCHEMA_NAME	VARCHAR2(30)		Schema to be renamed
TO_SCHEMA_NAME	VARCHAR2(30)		New schema name
FROM_TABLE_NAME	VARCHAR2(30)		Table to be renamed
TO_TABLE_NAME	VARCHAR2(30)		New table name
PRECEDENCE	NUMBER		4 (the execution order relative to other transformations on the same STEP_NUMBER; the smaller number will be executed first)
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_RULES

DBA_STREAMS_RULES displays information about the rules used by all Streams processes in the database. Its columns are the same as those in ALL_STREAMS_RULES.

See Also: "ALL_STREAMS_RULES" on page 3-55

DBA_STREAMS_SCHEMA_RULES

DBA_STREAMS_SCHEMA_RULES displays information about the schema rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_SCHEMA_RULES.

See Also: "ALL_STREAMS_SCHEMA_RULES" on page 3-56

DBA_STREAMS_TABLE_RULES

DBA_STREAMS_TABLE_RULES displays information about the table rules created for all Streams capture processes, propagations, and apply processes in the database. Its columns are the same as those in ALL_STREAMS_TABLE_RULES.

See Also: "ALL_STREAMS_TABLE_RULES" on page 3-57

DBA_STREAMS_TP_COMPONENT

DBA_STREAMS_TP_COMPONENT displays information about each Oracle Streams component at each database.

Column	Datatype	NULL	Description
COMPONENT_ID	NUMBER	NOT NULL	ID of the Streams component
COMPONENT_NAME	VARCHAR2(194)		Name of the Streams component
COMPONENT_DB	VARCHAR2(128)		Database where the Streams component resides

Column	Datatype	NULL	Description
COMPONENT_TYPE	VARCHAR2(20)		Type of the Streams component: ■ CAPTURE ■ PROPAGATION SENDER ■ PROPAGATION RECEIVER ■ APPLY ■ QUEUE
COMPONENT_CHANGED_TIME	DATE		Time that the Streams component was last changed

DBA_STREAMS_TP_COMPONENT_LINK

DBA_STREAMS_TP_COMPONENT_LINK displays information about how messages flow between Oracle Streams components.

Column	Datatype	NULL	Description
SOURCE_COMPONENT_ID	NUMBER	NOT NULL	ID of the source Streams component
SOURCE_COMPONENT_NAME	VARCHAR2(194)		Name of the source Streams component
SOURCE_COMPONENT_DB	VARCHAR2(128)		Database where the source Streams component resides
SOURCE_COMPONENT_TYPE	VARCHAR2(20)		Type of the source Streams component: ■ CAPTURE ■ PROPAGATION SENDER ■ PROPAGATION RECEIVER ■ APPLY ■ QUEUE
DESTINATION_COMPONENT_ID	NUMBER	NOT NULL	ID of the destination Streams component
DESTINATION_COMPONENT_NAME	VARCHAR2(194)		Name of the destination Streams component
DESTINATION_COMPONENT_DB	VARCHAR2(128)		Database where the destination Streams component resides
DESTINATION_COMPONENT_TYPE	VARCHAR2(20)		Type of the destination Streams component: ■ CAPTURE ■ PROPAGATION SENDER ■ PROPAGATION RECEIVER ■ APPLY ■ QUEUE
PATH_ID	NUMBER	NOT NULL	ID of the stream path
POSITION	NUMBER		Position of the link within the stream path
ACTIVE	VARCHAR2(3)		Indicates whether the Link is on an Active stream Path (YES) or not (NO)
ORIGINAL_PATH_ID	NUMBER		ID of the Original stream Path for the Optimized Path

DBA_STREAMS_TP_COMPONENT_STAT

DBA_STREAMS_TP_COMPONENT_STAT displays temporary performance statistics and session statistics about each Oracle Streams component.

Column	Datatype	NULL	Description
COMPONENT_ID	NUMBER	NOT NULL	ID of the Streams component

Column	Datatype	NULL	Description
COMPONENT_NAME	VARCHAR2(194)		Name of the Streams component
COMPONENT_DB	VARCHAR2(128)		Database where the Streams component resides
COMPONENT_TYPE	VARCHAR2(20)		Type of the Streams component: ▪ CAPTURE ▪ PROPAGATION SENDER ▪ PROPAGATION RECEIVER ▪ APPLY ▪ QUEUE
SUB_COMPONENT_TYPE	VARCHAR2(22)		Type of the Streams subcomponent: ▪ LOGMINER READER ▪ LOGMINER PREPARER ▪ LOGMINER BUILDER ▪ CAPTURE PROCESS ▪ APPLY NETWORK RECEIVER ▪ APPLY READER ▪ APPLY COORDINATOR ▪ APPLY SERVER
SESSION_ID	NUMBER		ID of the Streams session for the Streams component
SESSION_SERIAL#	NUMBER		Serial number of the Streams session for the Streams component
STATISTIC_TIME	DATE		Time that the statistic was taken
STATISTIC_NAME	VARCHAR2(64)		Name of the statistic
STATISTIC_VALUE	NUMBER		Value of the statistic
STATISTIC_UNIT	VARCHAR2(64)		Unit of the statistic
ADVISOR_RUN_ID	NUMBER		Logical number (1-based) of the Advisor run
ADVISOR_RUN_TIME	DATE		Time that the Advisor was run

DBA_STREAMS_TP_DATABASE

DBA_STREAMS_TP_DATABASE displays information about each database that contains Oracle Streams components.

Column	Datatype	NULL	Description
GLOBAL_NAME	VARCHAR2(128)	NOT NULL	Global name of the Streams database
LAST_QUERIED	DATE	NOT NULL	Time that the Streams database was last queried
VERSION	VARCHAR2(30)		Database version of the Streams database
COMPATIBILITY	VARCHAR2(30)		Compatible setting of the Streams database
MANAGEMENT_PACK_ ACCESS	VARCHAR2(30)		Management pack access of the Streams database

DBA_STREAMS_TP_PATH_BOTTLENECK

DBA_STREAMS_TP_PATH_BOTTLENECK displays temporary information about Oracle Streams components that might be slowing down the flow of messages in a stream path.

Column	Datatype	NULL	Description
PATH_ID	NUMBER		ID of the stream path

Column	Datatype	NULL	Description
COMPONENT_ID	NUMBER		ID of the bottleneck component
COMPONENT_NAME	VARCHAR2(194)		Name of the bottleneck component
COMPONENT_DB	VARCHAR2(128)		Database where the bottleneck component resides
COMPONENT_TYPE	VARCHAR2(20)		Type of the bottleneck component: ■ CAPTURE ■ PROPAGATION SENDER ■ PROPAGATION RECEIVER ■ APPLY ■ QUEUE
TOP_SESSION_ID	NUMBER		ID of the top session for the bottleneck component
TOP_SESSION_SERIAL#	NUMBER		Serial number of the top session for the bottleneck component
ACTION_NAME	VARCHAR2(32)		Action name for the bottleneck process
BOTTLENECK_IDENTIFIED	VARCHAR2(30)		Indicates whether the bottleneck was identified (YES) or not (NO)
ADVISOR_RUN_ID	NUMBER		Logical number (1-based) of the Advisor run
ADVISOR_RUN_TIME	DATE		Time that the Advisor was run
ADVISOR_RUN_REASON	VARCHAR2(4000)		Reason for the bottleneck analysis result.: ■ NULL - Bottleneck is identified ■ PRE-11.1 DATABASE EXISTS - A pre-release 11.1 database exists in the stream path ■ DIAGNOSTIC PACK REQUIRED - A database in the stream path does not have the diagnostic package installed ■ NO BOTTLENECK IDENTIFIED

DBA_STREAMS_TP_PATH_STAT

DBA_STREAMS_TP_PATH_STAT displays temporary performance statistics about each stream path that exists in the Oracle Streams topology.

Column	Datatype	NULL	Description
PATH_ID	NUMBER		ID of the stream path
STATISTIC_TIME	DATE		Time that the statistic was taken
STATISTIC_NAME	VARCHAR2(64)		Name of the statistic
STATISTIC_VALUE	NUMBER		Value of the statistic
STATISTIC_UNIT	VARCHAR2(64)		Unit of the statistic
ADVISOR_RUN_ID	NUMBER		Logical number (1-based) of the Advisor run
ADVISOR_RUN_TIME	DATE		Time that the Advisor was run

DBA_STREAMS_TRANSFORM_FUNCTION

DBA_STREAMS_TRANSFORM_FUNCTION displays information about all rule-based transformation functions in the database. Its columns are the same as those in ALL_STREAMS_TRANSFORM_FUNCTION.

See Also: "ALL_STREAMS_TRANSFORM_FUNCTION" on page 3-58

DBA_STREAMS_TRANSFORMATIONS

DBA_STREAMS_TRANSFORMATIONS displays information about all transformations available on a system, in order of execution.

Column	Datatype	NULL	Description
RULE_OWNER	VARCHAR2(30)		Owner of the rule
RULE_NAME	VARCHAR2(30)		Name of the rule
TRANSFORM_TYPE	VARCHAR2(26)		Type of the transformation: ■ DECLARATIVE TRANSFORMATION ■ SUBSET RULE ■ CUSTOM TRANSFORMATION
FROM_SCHEMA_NAME	VARCHAR2(30)		Schema to be renamed
TO_SCHEMA_NAME	VARCHAR2(30)		New schema name
FROM_TABLE_NAME	VARCHAR2(30)		Table to be renamed
TO_TABLE_NAME	VARCHAR2(30)		New table name
SCHEMA_NAME	VARCHAR2(30)		Schema of the column to be modified
TABLE_NAME	VARCHAR2(30)		Table of the column to be modified
FROM_COLUMN_NAME	VARCHAR2(4000)		Column to be renamed
TO_COLUMN_NAME	VARCHAR2(4000)		New column name
COLUMN_NAME	VARCHAR2(4000)		Column to add or delete
COLUMN_VALUE	ANYDATA		Value of the column to add
COLUMN_TYPE	VARCHAR2(4000)		Type of the new column
COLUMN_FUNCTION	VARCHAR2(30)		Name of the default function used to add a column
VALUE_TYPE	VARCHAR2(3)		Indicates whether to modify the old (OLD), new (NEW), or both (*) values of the LCR
USER_FUNCTION_NAME	VARCHAR2(4000)		Name of the user-defined transformation function to run
SUBSETTING_OPERATION	VARCHAR2(6)		DML operation for row subsetting: ■ INSERT ■ UPDATE ■ DELETE
DML_CONDITION	VARCHAR2(4000)		Row subsetting condition
DECLARATIVE_TYPE	VARCHAR2(13)		Type of declarative transform: ■ DELETE COLUMN ■ RENAME COLUMN ■ ADD COLUMN ■ RENAME TABLE ■ RENAME SCHEMA
PRECEDENCE	NUMBER		Execution order relative to other declarative transformations on the same STEP_NUMBER
STEP_NUMBER	NUMBER		Order in which this transformation should be executed

DBA_STREAMS_UNSUPPORTED

DBA_STREAMS_UNSUPPORTED displays information about all tables in the database that are not supported by Streams in this release of the Oracle Database. Its columns are the same as those in ALL_STREAMS_UNSUPPORTED.

See Also: "ALL_STREAMS_UNSUPPORTED" on page 3-59

DBA_SUBPART_COL_STATISTICS

DBA_SUBPART_COL_STATISTICS provides column statistics and histogram information for all subpartitions in the database. Its columns are the same as those in "ALL_STORED_SETTINGS" on page 3-51.

DBA_SUBPART_HISTOGRAMS

DBA_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on all table subpartitions in the database. Its columns are the same as those in "ALL_SUBPART_HISTOGRAMS" on page 3-61.

DBA_SUBPART_KEY_COLUMNS

DBA_SUBPART_KEY_COLUMNS lists subpartitioning key columns for all composite-partitioned tables (and local indexes on composite-partitioned tables) in the database. Its columns are the same as those in ALL_SUBPART_KEY_COLUMNS.

See Also: "ALL_SUBPART_KEY_COLUMNS" on page 3-61

DBA_SUBPARTITION_TEMPLATES

DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database. Its columns are the same as those in ALL_SUBPARTITION_TEMPLATES.

See Also: "ALL_SUBPARTITION_TEMPLATES" on page 3-62

DBA_SUBSCR_REGISTRATIONS

DBA_SUBSCR_REGISTRATIONS displays information about all subscription registrations in the database.

Related View

USER_SUBSCR_REGISTRATIONS displays information about the subscription registrations owned by the current user.

Column	Datatype	NULL	Description
REG_ID	NUMBER		Registration ID
SUBSCRIPTION_NAME	VARCHAR2(128)	NOT NULL	Name of the subscription registration. The subscription name is of the form *schema.queue* if the registration is for a single consumer queue or *schema.queue:consumer_name* if the registration is for a multi-consumer queue.
LOCATION_NAME	VARCHAR2(256)	NOT NULL	Location endpoint of the registration
USER#	NUMBER	NOT NULL	Internally generated user ID
USER_CONTEXT	RAW(128)		Context the user provided during registration of PL/SQL registrations or an internally generated context for OCI registrations
CONTEXT_SIZE	NUMBER		Size of the context
NAMESPACE	VARCHAR2(9)		Namespace of the subscription registration: ▪ ANONYMOUS ▪ AQ ▪ DBCHANGE

Column	Datatype	NULL	Description
PRESENTATION	VARCHAR2(7)		Presentation format of notifications: ■ DEFAULT - Binary ■ XML
VERSION	VARCHAR2(5)		Database version: ■ 8.1.6 ■ 10.2 ■ 11.1
STATUS	VARCHAR2(8)		Status of the registration: ■ DB REG - Database registration ■ LDAP REG - LDAP registration
ANY_CONTEXT	ANYDATA		AnyData user context (used for Streams replication registrations)
CONTEXT_TYPE	NUMBER		Type of the user context
QOSFLAGS	VARCHAR2(13)		Quality of service of the registration: ■ RELIABLE - Reliable notifications persist across instance and database restarts ■ PAYLOAD - Payload delivery is required. It is only supported for client notification and only for RAW queues. ■ PURGE_ON_NTFN - Registration is to be purged automatically when the first notification is delivered to this registration location
PAYLOAD_CALLBACK	VARCHAR2(4000)		Any callback registered to serialize the notification payload
TIMEOUT	TIMESTAMP(6)		Registration timeout
REG_TIME	TIMESTAMP(6) WITH TIME ZONE		Time of the registration
NTFN_GROUPING_CLASS	VARCHAR2(4)		Notification grouping class
NTFN_GROUPING_VALUE	NUMBER		Notification grouping value
NTFN_GROUPING_TYPE	VARCHAR2(7)		Notification grouping type: ■ SUMMARY ■ LAST
NTFN_GROUPING_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Notification grouping start time
NTFN_GROUPING_REPEAT_COUNT	VARCHAR2(40)		Notification grouping repeat count, or FOREVER

See Also: "USER_SUBSCR_REGISTRATIONS" on page 5-130

DBA_SUBSCRIBED_COLUMNS

DBA_SUBSCRIBED_COLUMNS describes the columns of source tables to which any subscriber has subscribed. This view is intended for use by Change Data Capture publishers.

Related View

USER_SUBSCRIBED_COLUMNS describes the columns of source tables to which the current user has subscribed.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Source table schema identifier
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Source table identifier
COLUMN_NAME	VARCHAR2(30)	NOT NULL	Source table column identifier
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription
SOURCE_DB	VARCHAR2(128)		Source database global name

See Also: "USER_SUBSCRIBED_COLUMNS" on page 5-130

DBA_SUBSCRIBED_TABLES

DBA_SUBSCRIBED_TABLES describes all source tables in the database to which any subscriber has subscribed. This view is intended for use by Change Data Capture publishers.

Related View

USER_SUBSCRIBED_TABLES describes the source tables to which the current user has subscribed.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handle
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Source table owner
SOURCE_TABLE_NAME	VARCHAR2(30)	NOT NULL	Source table name
VIEW_NAME	VARCHAR2(30)		Subscriber view name
CHANGE_SET_NAME	VARCHAR2(30)	NOT NULL	Change set name
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription

See Also: "USER_SUBSCRIBED_TABLES" on page 5-131

DBA_SUBSCRIPTIONS

DBA_SUBSCRIPTIONS describes all subscriptions in the database. This view is intended for use by Change Data Capture publishers.

Related View

USER_SUBSCRIPTIONS describes the subscriptions created by the current user.

Column	Datatype	NULL	Description
HANDLE	NUMBER	NOT NULL	Subscription handle
SET_NAME	VARCHAR2(30)	NOT NULL	Change set name
USERNAME	VARCHAR2(30)	NOT NULL	User name of the subscriber
CREATED	DATE	NOT NULL	Subscription creation date
STATUS	VARCHAR2(1)	NOT NULL	Status of the subscriptions: ■ N - Not yet active ■ A - Currently active
EARLIEST_SCN	NUMBER	NOT NULL	Subscription window low boundary

Column	Datatype	NULL	Description
LATEST_SCN	NUMBER	NOT NULL	Subscription window high boundary
DESCRIPTION	VARCHAR2(255)		Comment field for the subscriber
LAST_PURGED	DATE		Last time the subscriber called the PURGE_WINDOW for this subscription
LAST_EXTENDED	DATE		Last time the subscriber called the EXTEND_WINDOW for this subscription
SUBSCRIPTION_NAME	VARCHAR2(30)	NOT NULL	Name of the subscription

See Also: "USER_SUBSCRIPTIONS" on page 5-131

DBA_SYNC_CAPTURE

DBA_SYNC_CAPTURE displays information about all synchronous capture processes in the database. Its columns are the same as those in ALL_SYNC_CAPTURE.

See Also: "ALL_SYNC_CAPTURE" on page 3-63

DBA_SYNC_CAPTURE_PREPARED_TABS

DBA_SYNC_CAPTURE_PREPARED_TABS displays information about all tables in the database that are prepared for synchronous capture instantiation. Its columns are the same as those in ALL_SYNC_CAPTURE_PREPARED_TABS.

See Also: "ALL_SYNC_CAPTURE_PREPARED_TABS" on page 3-63

DBA_SYNC_CAPTURE_TABLES

DBA_SYNC_CAPTURE_TABLES displays information about all tables in the database that are captured by synchronous Streams captures. Its columns are the same as those in ALL_SYNC_CAPTURE_TABLES.

See Also: "ALL_SYNC_CAPTURE_TABLES" on page 3-64

DBA_SYNONYMS

DBA_SYNONYMS describes all synonyms in the database. Its columns are the same as those in ALL_SYNONYMS.

See Also: "ALL_SYNONYMS" on page 3-64

DBA_SYS_PRIVS

DBA_SYS_PRIVS describes system privileges granted to users and roles.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Grantee name, user, or role receiving the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN option (YES) or not (NO)

DBA_TAB_COL_STATISTICS

DBA_TAB_COL_STATISTICS contains column statistics and histogram information extracted from "DBA_TAB_COLUMNS" on page 5-56. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 3-65.

DBA_TAB_COLS

DBA_TAB_COLS describes the columns of all tables, views, and clusters in the database. This view differs from "DBA_TAB_COLUMNS" on page 5-56 in that hidden columns are not filtered out. Its columns are the same as those in "ALL_TAB_COLS" on page 3-66. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_TAB_COLUMNS

DBA_TAB_COLUMNS describes the columns of all tables, views, and clusters in the database. Its columns are the same as those in "ALL_TAB_COLUMNS" on page 3-68. To gather statistics for this view, use the SQL ANALYZE statement.

DBA_TAB_COMMENTS

DBA_TAB_COMMENTS displays comments on all tables and views in the database. Its columns are the same as those in ALL_TAB_COMMENTS.

See Also: "ALL_TAB_COMMENTS" on page 3-69

DBA_TAB_HISTGRM_PENDING_STATS

DBA_TAB_HISTGRM_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions in the database. Its columns are the same as those in "ALL_TAB_HISTGRM_PENDING_STATS" on page 3-70.

DBA_TAB_HISTOGRAMS

DBA_TAB_HISTOGRAMS describes histograms on columns of all tables in the database. Its columns are the same as those in "ALL_TAB_HISTOGRAMS" on page 3-70.

DBA_TAB_MODIFICATIONS

DBA_TAB_MODIFICATIONS describes modifications to all tables in the database that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in "ALL_TAB_MODIFICATIONS" on page 3-71.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time. For performance reasons, the Oracle Database does not populate this view immediately when the actual modifications occur. Run the FLUSH_DATABASE_MONITORING_INFO procedure in the DBMS_STATS PL/SQL package to populate this view with the latest information. The ANALYZE_ANY system privilege is required to run this procedure.

DBA_TAB_PARTITIONS

DBA_TAB_PARTITIONS provides the partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE statements for all partitions in the database. Its columns are the same as those in "ALL_TAB_PARTITIONS" on page 3-72.

DBA_TAB_PENDING_STATS

DBA_TAB_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions in the database. Its columns are the same as those in "ALL_TAB_PENDING_STATS" on page 3-73.

DBA_TAB_PRIVS

DBA_TAB_PRIVS describes all object grants in the database.

Related View

USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object. The object can be any object, including tables, packages, indexes, sequences, and so on.
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
PRIVILEGE	VARCHAR2(40)	NOT NULL	Privilege on the object
GRANTABLE	VARCHAR2(3)		Indicates whether the privilege was granted with the GRANT OPTION (YES) or not (NO)
HIERARCHY	VARCHAR2(3)		Indicates whether the privilege was granted with the HIERARCHY OPTION (YES) or not (NO)

See Also: "USER_TAB_PRIVS" on page 5-133

DBA_TAB_STATISTICS

DBA_TAB_STATISTICS displays optimizer statistics for all tables in the database. Its columns are the same as those in ALL_TAB_STATISTICS.

See Also: "ALL_TAB_STATISTICS" on page 3-75

DBA_TAB_STAT_PREFS

DBA_TAB_STAT_PREFS provides information about statistics preferences for all tables in the database. Its columns are the same as those in "ALL_TAB_STAT_PREFS" on page 3-75.

DBA_TAB_STATS_HISTORY

DBA_TAB_STATS_HISTORY provides a history of table statistics modifications for all tables in the database. Its columns are the same as those in "ALL_TAB_STATS_HISTORY" on page 3-76.

DBA_TAB_SUBPARTITIONS

DBA_TAB_SUBPARTITIONS describes, for each table subpartition, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in "ALL_TAB_SUBPARTITIONS" on page 3-77.

Note: Statistics are not collected on a per-subpartition basis.

DBA_TABLES

DBA_TABLES describes all relational tables in the database. Its columns are the same as those in ALL_TABLES. To gather statistics for this view, use the ANALYZE SQL statement.

See Also: "ALL_TABLES" on page 3-78

DBA_TABLESPACE_GROUPS

DBA_TABLESPACE_GROUPS describes all tablespace groups in the database.

Column	Datatype	NULL	Description
GROUP_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace group
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace

DBA_TABLESPACE_USAGE_METRICS

DBA_TABLESPACE_USAGE_METRICS describes tablespace usage metrics for all types of tablespaces, including permanent, temporary, and undo tablespaces.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name
USED_SPACE	NUMBER		Total space consumed by the tablespace
TABLESPACE_SIZE	NUMBER		Total size of the tablespace
USED_PERCENT	NUMBER		Percentage of used space, as a function of the maximum possible tablespace size

DBA_TABLESPACES

DBA_TABLESPACES describes all tablespaces in the database.

Related View

USER_TABLESPACES describes the tablespaces accessible to the current user. This view does not display the PLUGGED_IN column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
BLOCK_SIZE	NUMBER	NOT NULL	Tablespace block size
INITIAL_EXTENT	NUMBER		Default initial extent size
NEXT_EXTENT	NUMBER		Default incremental extent size
MIN_EXTENTS	NUMBER	NOT NULL	Default minimum number of extents
MAX_EXTENTS	NUMBER		Default maximum number of extents
MAX_SIZE	NUMBER		Default maximum size of segments
PCT_INCREASE	NUMBER		Default percent increase for extent size
MIN_EXTLEN	NUMBER		Minimum extent size for this tablespace
STATUS	VARCHAR2(9)		Tablespace status: ■ ONLINE ■ OFFLINE ■ READ ONLY
CONTENTS	VARCHAR2(9)		Tablespace contents: ■ UNDO ■ PERMANENT ■ TEMPORARY
LOGGING	VARCHAR2(9)		Default logging attribute: ■ LOGGING ■ NOLOGGING
FORCE_LOGGING	VARCHAR2(3)		Indicates whether the tablespace is under force logging mode (YES) or not (NO)
EXTENT_MANAGEMENT	VARCHAR2(10)		Indicates whether the extents in the tablespace are dictionary managed (DICTIONARY) or locally managed (LOCAL)
ALLOCATION_TYPE	VARCHAR2(9)		Type of extent allocation in effect for the tablespace: ■ SYSTEM ■ UNIFORM ■ USER
PLUGGED_IN	VARCHAR2(3)		Indicates whether the tablespace is plugged in (YES) or not (NO)
SEGMENT_SPACE_MANAGEMENT	VARCHAR2(6)		Indicates whether the free and used segment space in the tablespace is managed using free lists (MANUAL) or bitmaps (AUTO)
DEF_TAB_COMPRESSION	VARCHAR2(8)		Indicates whether default table compression is enabled (ENABLED) or not (DISABLED) **Note:** Enabling default table compression indicates that all tables in the tablespace will be created with table compression enabled unless otherwise specified.
RETENTION	VARCHAR2(11)		Undo tablespace retention: ■ GUARANTEE - Tablespace is an undo tablespace with RETENTION specified as GUARANTEE A RETENTION value of GUARANTEE indicates that unexpired undo in all undo segments in the undo tablespace should be retained even if it means that forward going operations that need to generate undo in those segments fail. ■ NOGUARANTEE - Tablespace is an undo tablespace with RETENTION specified as NOGUARANTEE ■ NOT APPLY - Tablespace is not an undo tablespace

Column	Datatype	NULL	Description
BIGFILE	VARCHAR2(3)		Indicates whether the tablespace is a bigfile tablespace (YES) or a smallfile tablespace (NO)
PREDICATE_EVALUATION	VARCHAR2(7)		Indicates whether predicates are evaluated by host (HOST) or by storage (STORAGE)
ENCRYPTED	VARCHAR2(3)		Indicates whether the tablespace is encrypted (YES) or not (NO)

See Also: "USER_TABLESPACES" on page 5-134

DBA_TEMP_FILES

DBA_TEMP_FILES describes all temporary files (tempfiles) in the database.

Column	Datatype	NULL	Description
FILE_NAME	VARCHAR2(513)		Name of the database temp file
FILE_ID	NUMBER		File identifier number of the database temp file
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace to which the file belongs
BYTES	NUMBER		Size of the file (in bytes)
BLOCKS	NUMBER		Size of the file (in Oracle blocks)
STATUS	CHAR(9)		File status: ■ AVAILABLE
RELATIVE_FNO	NUMBER		Tablespace-relative file number
AUTOEXTENSIBLE	VARCHAR2(3)		Indicates whether the file is autoextensible (YES) or not (NO)
MAXBYTES	NUMBER		maximum size of the file (in bytes)
MAXBLOCKS	NUMBER		Maximum size of the file (in Oracle blocks)
INCREMENT_BY	NUMBER		Default increment for autoextension
USER_BYTES	NUMBER		Size of the useful portion of the file (in bytes)
USER_BLOCKS	NUMBER		Size of the useful portion of the file (in Oracle blocks)

DBA_TEMP_FREE_SPACE

DBA_TEMP_FREE_SPACE displays temporary space usage information at tablespace level.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace
TABLESPACE_SIZE	NUMBER		Total size of the tablespace, in bytes
ALLOCATED_SPACE	NUMBER		Total allocated space, in bytes, including space that is currently allocated and used and space that is currently allocated and available for reuse
FREE_SPACE	NUMBER		Total free space available, in bytes, including space that is currently allocated and available for reuse and space that is currently unallocated

DBA_THRESHOLDS

DBA_THRESHOLDS describes all thresholds.

Column	Datatype	NULL	Description
METRICS_NAME	VARCHAR2(64)		Metrics name
WARNING_OPERATOR	VARCHAR2(12)		Relational operator for warning thresholds: ■ GT ■ EQ ■ LT ■ LE ■ GE ■ CONTAINS ■ NE ■ DO NOT CHECK ■ DO_NOT_CHECK
WARNING_VALUE	VARCHAR2(256)		Warning threshold value
CRITICAL_OPERATOR	VARCHAR2(12)		Relational operator for critical thresholds: ■ GT ■ EQ ■ LT ■ LE ■ GE ■ CONTAINS ■ NE ■ DO NOT CHECK ■ DO_NOT_CHECK
CRITICAL_VALUE	VARCHAR2(256)		Critical threshold value
OBSERVATION_PERIOD	NUMBER		Observation period length (in minutes)
CONSECUTIVE_ OCCURRENCES	NUMBER		Number of occurrences before an alert is issued
INSTANCE_NAME	VARCHAR2(16)		Instance name; null for database-wide alerts
OBJECT_TYPE	VARCHAR2(64)		Object type: ■ SYSTEM ■ SERVICE ■ EVENT_CLASS ■ TABLESPACE ■ FILE
OBJECT_NAME	VARCHAR2(513)		Name of the object for which the threshold is set
STATUS	VARCHAR2(7)		Indicates whether the threshold is applicable on a valid object (VALID) or not (INVALID)

DBA_TRANSFORMATIONS

DBA_TRANSFORMATIONS displays information about all transformations in the database. These transformations can be specified with Advanced Queuing operations such as enqueue, dequeue, and subscribe to automatically integrate transformations in AQ messaging.

Related View

USER_TRANSFORMATIONS displays information about transformations owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
TRANSFORMATION_ID	NUMBER	NOT NULL	Unique identifier for the transformation
OWNER	VARCHAR2(30)	NOT NULL	Owning user of the transformation
NAME	VARCHAR2(30)	NOT NULL	Transformation name
FROM_TYPE	VARCHAR2(61)		Source type name
TO_TYPE	VARCHAR2(91)		Target type name

See Also: "USER_TRANSFORMATIONS" on page 5-134

DBA_TRIGGER_COLS

DBA_TRIGGER_COLS describes the use of columns in all triggers in the database. Its columns are the same as those in ALL_TRIGGER_COLS.

See Also: "ALL_TRIGGER_COLS" on page 3-80

DBA_TRIGGER_ORDERING

DBA_TRIGGER_ORDERING describes all triggers in the database that have FOLLOWS or PRECEDES ordering. Its columns are the same as those in ALL_TRIGGER_ORDERING.

See Also: "ALL_TRIGGER_ORDERING" on page 3-81

DBA_TRIGGERS

DBA_TRIGGERS describes all triggers in the database. Its columns are the same as those in ALL_TRIGGERS.

See Also: "ALL_TRIGGERS" on page 3-82

DBA_TS_QUOTAS

DBA_TS_QUOTAS describes tablespace quotas for all users.

Related View

USER_TS_QUOTAS describes tablespace quotas for the current user. This view does not display the USERNAME column.

Column	Datatype	NULL	Description
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Tablespace name
USERNAME	VARCHAR2(30)	NOT NULL	User with resource rights on the tablespace
BYTES	NUMBER		Number of bytes charged to the user
MAX_BYTES	NUMBER		User's quota in bytes, or -1 if no limit
BLOCKS	NUMBER	NOT NULL	Number of Oracle blocks charged to the user
MAX_BLOCKS	NUMBER		User's quota in Oracle blocks, or -1 if no limit
DROPPED	VARCHAR2(3)		Whether the tablespace has been dropped

DBA_TSM_DESTINATION

DBA_TSM_DESTINATION lists transparent session migration (TSM) destination session statistics.

Column	Datatype	NULL	Description
SOURCE_DATABASE_NAME	VARCHAR2(4000)		Database name of source session
DESTINATION_DATABASE_NAME	VARCHAR2(4000)		Database name of destination session
DESTINATION_INSTANCE_NAME	VARCHAR2(4000)		Instance name of destination session
DESTINATION_INSTANCE_ID	VARCHAR2(4000)		Instance ID of destination session
DESTINATION_INST_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Instance start time of destination session
SEQUENCE#	NUMBER		Migration sequence number
DESTINATION_SID	NUMBER		Session ID of destination session
DESTINATION_SERIAL#	NUMBER		Session serial number of destination session
DESTINATION_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Start time for migration on destination session
DESTINATION_END_TIME	TIMESTAMP(6) WITH TIME ZONE		End time for migration on destination session
DESTINATION_USER_NAME	VARCHAR2(30)	NOT NULL	User associated with the destination session
DESTINATION_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema associated with the destination session
DESTINATION_STATE	VARCHAR2(24)		Migration state of destination session

DBA_TSM_SOURCE

DBA_TSM_SOURCE lists transparent session migration (TSM) source session statistics.

Column	Datatype	NULL	Description
SOURCE_DATABASE_NAME	VARCHAR2(4000)		Database name of source session
SOURCE_INSTANCE_NAME	VARCHAR2(4000)		Instance name of source session
SOURCE_INSTANCE_ID	VARCHAR2(4000)		Instance ID of source session
SOURCE_INSTANCE_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Instance start time of source session
SEQUENCE#	NUMBER		Migration sequence number
SOURCE_SID	NUMBER		Session ID of source session
SOURCE_SERIAL#	NUMBER		Source serial number of source session
SOURCE_STATE	VARCHAR2(24)		Migration state of source session
CONNECT_STRING	VARCHAR2(4000)		Connect string specified for migration
SOURCE_START_TIME	TIMESTAMP(6) WITH TIME ZONE		Start time for migration on source session
COST	NUMBER		Estimate of migration cost
FAILURE_REASON	VARCHAR2(34)		Reason for migration failure, if any
SOURCE_END_TIME	TIMESTAMP(6) WITH TIME ZONE		End time for migration on source session
ROUNDTRIPS	NUMBER		Number of client-server round trips during migration
SOURCE_USER_NAME	VARCHAR2(30)	NOT NULL	User associated with the source session

Column	Datatype	NULL	Description
SOURCE_SCHEMA_NAME	VARCHAR2(30)	NOT NULL	Schema associated with the source session
DESTINATION_DATABASE_NAME	VARCHAR2(4000)		Database name of the destination session

DBA_TUNE_MVIEW

DBA_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure.

Related View

USER_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the task
TASK_NAME	VARCHAR2(30)		Name of the task
ACTION_ID	NUMBER	NOT NULL	Identifier of the action
SCRIPT_TYPE	VARCHAR2(14)		Type of the script: ■ IMPLEMENTATION ■ UNDO
STATEMENT	CLOB		Action statement

See Also: "USER_TUNE_MVIEW" on page 5-135

DBA_TYPE_ATTRS

DBA_TYPE_ATTRS describes the attributes of all object types in the database. Its columns are the same as those in ALL_TYPE_ATTRS.

See Also: "ALL_TYPE_ATTRS" on page 3-83

DBA_TYPE_METHODS

DBA_TYPE_METHODS describes the methods of all object types in the database. Its columns are the same as those in ALL_TYPE_METHODS.

See Also: "ALL_TYPE_METHODS" on page 3-84

DBA_TYPE_VERSIONS

DBA_TYPE_VERSIONS describes the versions of all object types in the database. Its columns are the same as those in ALL_TYPE_VERSIONS.

See Also: "ALL_TYPE_VERSIONS" on page 3-84

DBA_TYPES

DBA_TYPES describes all object types in the database. Its columns are the same as those in ALL_TYPES.

See Also: "ALL_TYPES" on page 3-85

DBA_UNDO_EXTENTS

DBA_UNDO_EXTENTS describes the extents comprising the segments in all undo tablespaces in the database.

Column	Datatype	NULL	Description
OWNER	CHAR(3)		Owner of the undo tablespace
SEGMENT_NAME	VARCHAR2(30)	NOT NULL	Name of the undo segment
TABLESPACE_NAME	VARCHAR2(30)	NOT NULL	Name of the undo tablespace
EXTENT_ID	NUMBER		ID of the extent
FILE_ID	NUMBER	NOT NULL	File identifier number of the file containing the extent
BLOCK_ID	NUMBER		Start block number of the extent
BYTES	NUMBER		Size of the extent (in bytes)
BLOCKS	NUMBER		Size of the extent (in blocks)
RELATIVE_FNO	NUMBER		Relative number of the file containing the segment header
COMMIT_JTIME	NUMBER		Commit time of the undo in the extent expressed as Julian time. This column is deprecated, but retained for backward compatibility reasons.
COMMIT_WTIME	VARCHAR2(20)		Commit time of the undo in the extent expressed as Wallclock time.This column is deprecated, but retained for backward compatibility reasons
STATUS	VARCHAR2(9)		Transaction Status of the undo in the extent: ▪ ACTIVE ▪ EXPIRED ▪ UNEXPIRED

DBA_UNUSED_COL_TABS

DBA_UNUSED_COL_TABS describes all tables in the database containing unused columns. Its columns are the same as those in ALL_UNUSED_COL_TABS.

See Also: "ALL_UNUSED_COL_TABS" on page 3-86

DBA_UPDATABLE_COLUMNS

DBA_UPDATABLE_COLUMNS describes all columns in a join view that can be updated by the database administrator, subject to appropriate privileges. Its columns are the same as those in "ALL_UPDATABLE_COLUMNS" on page 3-86.

See Also: *Oracle Database Concepts* for information on updatable join views

DBA_USERS

DBA_USERS describes all users of the database.

Related View

USER_USERS describes the current user. This view does not display the PASSWORD, PROFILE, PASSWORD_VERSIONS, or EDITIONS_ENABLED columns.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user
USER_ID	NUMBER	NOT NULL	ID number of the user
PASSWORD	VARCHAR2(30)		Encrypted password
ACCOUNT_STATUS	VARCHAR2(32)	NOT NULL	Account status: ■ OPEN ■ EXPIRED ■ EXPIRED(GRACE) ■ LOCKED(TIMED) ■ LOCKED ■ EXPIRED & LOCKED(TIMED) ■ EXPIRED(GRACE) & LOCKED(TIMED) ■ EXPIRED & LOCKED ■ EXPIRED(GRACE) & LOCKED
LOCK_DATE	DATE		Date the account was locked if account status was LOCKED
EXPIRY_DATE	DATE		Date of expiration of the account
DEFAULT_TABLESPACE	VARCHAR2(30)	NOT NULL	Default tablespace for data
TEMPORARY_TABLESPACE	VARCHAR2(30)	NOT NULL	Name of the default tablespace for temporary tables or the name of a tablespace group
CREATED	DATE	NOT NULL	User creation date
PROFILE	VARCHAR2(30)	NOT NULL	User resource profile name
INITIAL_RSRC_CONSUMER_GROUP	VARCHAR2(30)		Initial resource consumer group for the user
EXTERNAL_NAME	VARCHAR2(4000)		User external name
PASSWORD_VERSIONS	VARCHAR2(8)		Database version in which the password was created or changed
EDITIONS_ENABLED	VARCHAR2(1)		Reserved for future use

See Also: "USER_USERS" on page 5-136

DBA_USERS_WITH_DEFPWD

DBA_USERS_WITH_DEFPWD displays all users in the database that are still using their default passwords.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)	NOT NULL	Name of the user

DBA_USTATS

DBA_USTATS describes the user-defined statistics collected on all tables and indexes in the database. Its columns are the same as those in ALL_USTATS.

See Also: "ALL_USTATS" on page 3-87

DBA_VARRAYS

DBA_VARRAYS describes all varrays in the database. Its columns are the same as those in ALL_VARRAYS.

See Also: "ALL_VARRAYS" on page 3-88

DBA_VIEWS

DBA_VIEWS describes all views in the database. Its columns are the same as those in ALL_VIEWS.

See Also: "ALL_VIEWS" on page 3-88

DBA_WAITERS

DBA_WAITERS shows all the sessions that are waiting for a lock.

Column	Datatype	NULL	Description
WAITING_SESSION	NUMBER		The waiting session
HOLDING_SESSION	NUMBER		The holding session
LOCK_TYPE	VARCHAR2(26)		The lock type
MODE_HELD	VARCHAR2(40)		The mode held
MODE_REQUESTED	VARCHAR2(40)		The mode requested
LOCK_ID1	VARCHAR2(40)		Lock ID 1
LOCK_ID2	VARCHAR2(40)		Lock ID 2

DBA_WARNING_SETTINGS

DBA_WARNING_SETTINGS displays information about the warning parameter settings for all objects in the database. Its columns are the same as those in ALL_WARNING_SETTINGS.

See Also: "ALL_WARNING_SETTINGS" on page 3-89

DBA_WORKLOAD_CAPTURES

DBA_WORKLOAD_CAPTURES displays all the workload captures that have been performed in the current database. It also lists captures on which DBMS_WORKLOAD_CAPTURE.GET_CAPTURE_INFO() or DBMS_WORKLOAD_REPLAY.GET_REPLAY_INFO() have been called. Each row contains information about one workload capture.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Internal key for the workload capture
NAME	VARCHAR2(100)	NOT NULL	Name for the workload capture
DBID	NUMBER	NOT NULL	ID of the database in which the workload was captured
DBNAME	VARCHAR2(10)	NOT NULL	Name of the database in which the workload was captured
DBVERSION	VARCHAR2(30)	NOT NULL	Version of the database in which the workload was captured
PARALLEL	VARCHAR2(3)		Indicates whether the database in which the workload was captured is a RAC database (YES) or a single instance database (NO)
DIRECTORY	VARCHAR2(30)	NOT NULL	Name of the directory object for workload capture

Column	Datatype	NULL	Description
STATUS	VARCHAR2(40)	NOT NULL	Current status of the workload capture: ■ IN PROGRESS - Workload capture is in progress ■ COMPLETED - Workload capture has completed successfully ■ FAILED - Workload capture was aborted due to errors encountered (see the COMMENT column for further details)
START_TIME	DATE	NOT NULL	Datetime when the capture began
END_TIME	DATE		Datetime when the capture completed or failed; NULL if the capture is still in progress
DURATION_SECS	NUMBER		Duration of the workload capture (in seconds)
START_SCN	NUMBER	NOT NULL	Start SCN value for this capture
END_SCN	NUMBER		End SCN value for this capture; NULL if the capture is still in progress
DEFAULT_ACTION	VARCHAR2(30)	NOT NULL	Mode in which to apply workload capture filters: ■ INCLUDE - All the capture filters are treated as EXCLUSION filters, determining the workload that will not be captured. ■ EXCLUDE - All the capture filters are treated as INCLUSION FILTERS, determining the workload that will be captured.
FILTERS_USED	NUMBER		Number of filters that were used for this capture
CAPTURE_SIZE	NUMBER		Total size of workload capture
DBTIME	NUMBER		Total amount of database time (in microseconds) that has been recorded in this workload capture
DBTIME_TOTAL	NUMBER		Total amount of database time (in microseconds) across the entire database during the workload capture, including the part of the workload that was not captured.
USER_CALLS	NUMBER		Total number of user calls that have been recorded in this workload capture
USER_CALLS_TOTAL	NUMBER		Total number of user calls across the entire database during the workload capture, including the part of the workload that was not captured.
USER_CALLS_UNREPLAYABLE	NUMBER		Total number of user calls that will not be replayed in a subsequent replay of this workload capture
TRANSACTIONS	NUMBER		Total number of transactions that have been recorded in this workload capture
TRANSACTIONS_TOTAL	NUMBER		Total number of transactions across the entire database during the workload capture, including the part of the workload that was not captured.
CONNECTS	NUMBER		Total number of session connects that have been recorded in this workload capture
CONNECTS_TOTAL	NUMBER		Total number of session connects across the entire database during the workload capture, including the part of the workload that was not captured
ERRORS	NUMBER		Total number of errors that have been recorded in this workload capture
AWR_DBID	NUMBER		Database ID of the AWR snapshots that correspond to this workload capture. For captures that were performed in the current database, this value is equal to the current database's DBID. For captures that were performed in other databases, this value will either be NULL or will be populated by DBMS_WORKLOAD_CAPTURE.IMPORT_AWR().

Column	Datatype	NULL	Description
AWR_BEGIN_SNAP	NUMBER		Begin snapshot ID of the AWR snapshots that correspond to this workload capture
AWR_END_SNAP	NUMBER		End snapshot ID of the AWR snapshots that correspond to this workload capture
AWR_EXPORTED	VARCHAR2(12)		Indicates whether the AWR snapshots that correspond to this workload capture have been exported using DBMS_WORKLOAD_CAPTURE.EXPORT_AWR() (YES) or not (NO), or whether AWR snapshots cannot be exported because the capture is still in progress, has run to completion successfully, or was done in a different database from which it was not exported (NOT POSSIBLE)
ERROR_CODE	NUMBER		Error code for this workload capture
ERROR_MESSAGE	VARCHAR2(300)		Error message for this workload capture
DIR_PATH	VARCHAR2(4000)	NOT NULL	Full directory path for the workload capture directory object
DIR_PATH_SHARED	VARCHAR2(10)	NOT NULL	Indicates whether the workload capture directory is shared by all the instances of the recording database (applicable only for RAC databases)
LAST_PROCESSED_VERSION	VARCHAR2(30)		Database version in which this capture was preprocessed using DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE() last; NULL if the capture has never been preprocessed

DBA_WORKLOAD_CONNECTION_MAP

DBA_WORKLOAD_CONNECTION_MAP displays the connection mapping information for workload replay. Each row defines one connection mapping for a particular workload replay.

Column	Datatype	NULL	Description
REPLAY_ID	NUMBER	NOT NULL	ID of the replay (corresponds to DBA_WORKLOAD_REPLAYS.ID)
CONN_ID	NUMBER	NOT NULL	Key (ID) of the connection mapping table
CAPTURE_CONN	VARCHAR2(4000)	NOT NULL	Connection string that was used during capture
REPLAY_CONN	VARCHAR2(4000)		Connection string that should be used during replay

DBA_WORKLOAD_FILTERS

DBA_WORKLOAD_FILTERS displays all the workload filters that have been defined in the current database. In Oracle Database 11*g*, only workload filters of type CAPTURE are supported.

Column	Datatype	NULL	Description
TYPE	VARCHAR2(30)	NOT NULL	Type of the workload filter
ID	NUMBER	NOT NULL	Sequence number of the workload filter

Column	Datatype	NULL	Description
STATUS	VARCHAR2(6)		Status of the workload filter: ■ NEW - This filter will be used by the next subsequent operation such as the next workload capture. ■ IN USE - This filter is currently being used by an operation that is in progress such as an active workload capture. ■ USED - This filter was used in the past by some operation such as a past workload capture.
NAME	VARCHAR2(100)	NOT NULL	Name of the workload filter
ATTRIBUTE	VARCHAR2(100)	NOT NULL	Name of the attribute on which the filter is defined
VALUE	VARCHAR2(4000)	NOT NULL	Value of the attribute on which the filter is defined. Wildcards such as % and _ are supported if the attribute is of string type.

DBA_WORKLOAD_REPLAY_DIVERGENCE

DBA_WORKLOAD_REPLAY_DIVERGENCE displays information about data/error divergence for a user call that has been replayed.

Column	Datatype	NULL	Description
REPLAY_ID	NUMBER	NOT NULL	ID (key) for the workload replay
TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE		Time that the divergence occurred
DIVERGENCE_TYPE	NUMBER	NOT NULL	Reserved for future use
IS_QUERY_DATA_DIVERGENCE	VARCHAR2(1)		Indicates whether the data divergence is from the number of rows fetched by SELECT queries (Y) or not (N)
IS_DML_DATA_DIVERGENCE	VARCHAR2(1)		Indicates whether the divergence is from the number of rows affected by INSERT, UPDATE, or DELETE SQL statements (Y) or not (N)
IS_ERROR_DIVERGENCE	VARCHAR2(1)		Indicates whether the divergence is from errors seen during capture or replay (Y) or not (N)
IS_THREAD_FAILURE	VARCHAR2(1)		Indicates whether the divergence is from sessions that failed during replay (Y) or not (N)
EXPECTED_ROW_COUNT	NUMBER		Number of rows fetched for SELECT queries or rows affected for INSERT, UPDATE, or DELETE SQL statements during capture
OBSERVED_ROW_COUNT	NUMBER		Actual number of rows fetched for SELECT queries or rows affected for INSERT, UPDATE, or DELETE SQL statements during replay
EXPECTED_ERROR#	NUMBER		Error number that was seen during capture (0 if the capture ran successfully)
OBSERVED_ERROR#	NUMBER		Actual error number seen during replay (0 if the replay ran successfully, 15566 (corresponding to ORA-15566) if the captured call could not be replayed)
STREAM_ID	NUMBER	NOT NULL	Stream ID of the session that reported the divergence
CALL_COUNTER	NUMBER	NOT NULL	Call counter of the user call that reported the divergence
SQL_ID	VARCHAR2(13)		SQL ID of the SQL that reported the divergence
SESSION_ID	NUMBER	NOT NULL	Session ID of the session that reported the divergence
SESSION_SERIAL#	NUMBER	NOT NULL	Captured session serial number of the session that reported the divergence
SERVICE	VARCHAR2(64)		Service name of the session that reported the divergence

Column	Datatype	NULL	Description
MODULE	VARCHAR2(48)		Module name of the session that reported the divergence
ACTION	VARCHAR2(32)		Action name of the session that reported the divergence

DBA_WORKLOAD_REPLAYS

DBA_WORKLOAD_REPLAYS displays all the workload replays that have been performed in the current database. It also lists replays on which DBMS_WORKLOAD_REPLAY.GET_REPLAY_INFO() has been called. Each row contains information about one workload replay.

Column	Datatype	NULL	Description
ID	NUMBER	NOT NULL	Internal key for the workload replay
NAME	VARCHAR2(100)	NOT NULL	Name of the workload replay
DBID	NUMBER	NOT NULL	ID of the database in which the workload was replayed
DBNAME	VARCHAR2(10)	NOT NULL	Name of the database in which the workload was replayed
DBVERSION	VARCHAR2(30)	NOT NULL	Version of the database in which the workload was replayed
PARALLEL	VARCHAR2(3)		Indicates whether the database in which the workload was replayed was a RAC database (YES) or a single instance database (NO)
DIRECTORY	VARCHAR2(30)	NOT NULL	Name of the directory object for the workload replay
CAPTURE_ID	NUMBER		ID of the capture (DBA_WORKLOAD_CAPTURES.ID) that was replayed
STATUS	VARCHAR2(40)	NOT NULL	Current status of the workload replay: ■ PREPARE - Workload prepare has been started and is waiting for clients to join ■ IN PROGRESS - Workload replay is in progress ■ COMPLETED - Workload replay has successfully completed ■ CANCELLED - Workload replay or the workload prepare has been cancelled ■ FAILED - Workload replay was aborted due to errors encountered. See the COMMENTS column for further information.
PREPARE_TIME	DATE		Datetime at which the workload prepare started
START_TIME	DATE		Datetime when the replay began
END_TIME	DATE		Datetime when the replay completed or cancelled; NULL if the replay is still in progress
DURATION_SECS	NUMBER		Duration of the workload replay (in seconds)
NUM_CLIENTS	NUMBER	NOT NULL	Number of workload replay client processes that were used in this workload replay
NUM_CLIENTS_DONE	NUMBER	NOT NULL	Number of workload replay client processes that have finished replay
DEFAULT_ACTION	VARCHAR2(30)	NOT NULL	Reserved for future use

Column	Datatype	NULL	Description
SYNCHRONIZATION	VARCHAR2(5)		Indicates whether recorded transaction semantics should be maintained (TRUE) or not (FALSE) When synchronization is on, the commit order observed during the original workload capture will be preserved. Every action that is replayed will be executed only after all of its dependent commits have been executed. Dependent commits are commits that were issued before the given action in the original workload capture. **See Also:** DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY() in *Oracle Database PL/SQL Packages and Types Reference* for a detailed explanation of this replay parameter
CONNECT_TIME_SCALE	NUMBER	NOT NULL	Connection time scaling factor for captured streams during replay. The value is interpreted as a percentage.The default value of 100 means 100 percent. **See Also:** DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY() in *Oracle Database PL/SQL Packages and Types Reference* for a detailed explanation of this replay parameter
THINK_TIME_SCALE	NUMBER	NOT NULL	Think time scaling factor for captured streams during replay. It scales the thinking time elapsed between two successive user calls from the same captured stream. The input is interpreted as a percentage. The default value of 100 means 100 percent. **See Also:** DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY() in *Oracle Database PL/SQL Packages and Types Reference* for a detailed explanation of this replay parameter
THINK_TIME_AUTO_CORRECT	VARCHAR2(5)		Indicates whether the think time should be automatically corrected between calls (TRUE) or not (FALSE) A value of TRUE either reduces think time if replay goes slower than capture or increases think time if replay goes faster than capture. A value of FALSE results in no action. **See Also:** DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY() in *Oracle Database PL/SQL Packages and Types Reference* for a detailed explanation of this replay parameter
USER_CALLS	NUMBER		Total number of user calls replayed
DBTIME	NUMBER		Accumulated database time for the replay
NETWORK_TIME	NUMBER		Accumulated network time for the replay
THINK_TIME	NUMBER		Accumulated think time for the replay
ELAPSED_TIME_DIFF	NUMBER		Reserved for future use
NEW_ERRORS	NUMBER		Total number of new errors
MUTATED_ERRORS	NUMBER		Total number of different errors
AWR_DBID	NUMBER		Database ID of the AWR snapshots that correspond to this workload replay. For replays that were performed in the current database, this value is equal to the current database's DBID. For replays that were performed in other databases, this value will either be NULL or will be populated by DBMS_WORKLOAD_REPLAY.IMPORT_AWR().
AWR_BEGIN_SNAP	NUMBER		Begin snapshot ID of the AWR snapshots that correspond to this workload replay
AWR_END_SNAP	NUMBER		End snapshot ID of the AWR snapshots that correspond to this workload replay

Column	Datatype	NULL	Description
AWR_EXPORTED	VARCHAR2(12)		Indicates whether the AWR snapshots that correspond to this workload replay have been exported using DBMS_WORKLOAD_REPLAY.EXPORT_AWR() (YES) or not (NO), or whether AWR snapshots cannot be exported because the replay is still in progress, has run to completion successfully, or was done in a different database from which it was not exported (NOT POSSIBLE)
ERROR_CODE	NUMBER		Error code for this workload replay
ERROR_MESSAGE	VARCHAR2(300)		Error message for this workload replay
DIR_PATH	VARCHAR2(4000)	NOT NULL	Full directory path for the replay directory object

DBA_XML_INDEXES

DBA_XML_INDEXES describes all XML indexes in the database. Its columns are the same as those in ALL_XML_INDEXES.

See Also: "ALL_XML_INDEXES" on page 3-90

DBA_XML_SCHEMAS

DBA_XML_SCHEMAS describes all registered XML schemas in the database. Its columns are the same as those in ALL_XML_SCHEMAS.

See Also: "ALL_XML_SCHEMAS" on page 3-91

DBA_XML_TAB_COLS

DBA_XML_TAB_COLS describes the columns of all XML tables in the database. Its columns are the same as those in ALL_XML_TAB_COLS.

See Also: "ALL_XML_TAB_COLS" on page 3-91

DBA_XML_TABLES

DBA_XML_TABLES describes all XML tables in the database. Its columns are the same as those in ALL_XML_TABLES.

See Also: "ALL_XML_TABLES" on page 3-92

DBA_XML_VIEW_COLS

DBA_XML_VIEW_COLS describes the columns of all XML views in the database. Its columns are the same as those in ALL_XML_VIEW_COLS.

See Also: "ALL_XML_VIEW_COLS" on page 3-93

DBA_XML_VIEWS

DBA_XML_VIEWS describes all XML views in the database. Its columns are the same as those in ALL_XML_VIEWS.

See Also: "ALL_XML_VIEWS" on page 3-93

DBMS_ALERT_INFO

DBMS_ALERT_INFO describes registered alerts.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	Name of the alert
SID	VARCHAR2(30)	NOT NULL	Session ID of a session waiting for this alert
CHANGED	VARCHAR2(1)		Boolean flag to indicate that an alert has been signaled. Y: alert signaled, N: no alert.
MESSAGE	VARCHAR2(1800)		Optional message passed by signaler

DBMS_LOCK_ALLOCATED

DBMS_LOCK_ALLOCATED describes user-allocated locks.

Column	Datatype	NULL	Description
NAME	VARCHAR2(128)	NOT NULL	Name of the lock
LOCKID	NUMBER(38)		Lock identifier number
EXPIRATION	DATE		Planned lock expiration date (updates whenever the allocation procedure is run)

DEPTREE

This view, created by utldtree.sql, contains information on the object dependency tree. For user SYS, this view displays shared cursors (and only shared cursors) that depend on the object. For all other users, it displays objects other than shared cursors. Other users can access SYS.DEPTREE for information on shared cursors.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name
SEQ#	NUMBER		Sequence number in the dependency tree. Used for ordering queries. **See Also:** "IDEPTREE" on page 5-81

DICT

DICT is a synonym for DICTIONARY.

See Also: "DICTIONARY" on page 5-75

DICT_COLUMNS

DICT_COLUMNS contains descriptions of columns in data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object that contains the column
COLUMN_NAME	VARCHAR2(30)		Name of the column
COMMENTS	VARCHAR2(4000)		Text comment on the column

DICTIONARY

DICTIONARY contains descriptions of data dictionary tables and views.

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)		Name of the object
COMMENTS	VARCHAR2(4000)		Text comment on the object

DM_USER_MODELS

DM_USER_MODELS ???.

Column	Datatype	NULL	Description
NAME	VARCHAR2(30)	NOT NULL	???
FUNCTION_NAME	VARCHAR2(30)		???
ALGORITHM_NAME	VARCHAR2(30)		???
CREATION_DATE	DATE	NOT NULL	???
BUILD_DURATION	NUMBER		???
TARGET_ATTRIBUTE	VARCHAR2(30)		???
MODEL_SIZE	NUMBER		???

DOCUMENT_LINKS

DOCUMENT_LINKS provides system information about Oracle XML DB document links in Oracle XML DB Repository documents. When an XML document that includes XLink or XInclude links is added to the repository, these links can be mapped to document links, which are tracked using view DOCUMENT_LINKS.

Column	Datatype	NULL	Description
SOURCE_ID	RAW(16)		The source resource OID
TARGET_ID	RAW(16)		The target resource OID
TARGET_PATH	VARCHAR2(4000)		This column is always NULL. It is reserved for future use.
LINK_TYPE	VARCHAR2(8)		The document link type: Hard or Weak
LINK_FORM	VARCHAR2(8)		Whether the original link was of form XLink or XInclude
SOURCE_TYPE	VARCHAR2(17)		Whether the link is contained in Resource Content or Resource Metadata

See Also: *Oracle XML DB Developer's Guide* for information about using this view

ERROR_SIZE

ERROR_SIZE is accessed to create "DBA_OBJECT_SIZE" on page 5-3 and "USER_OBJECT_SIZE" on page 5-119.

EXCEPTIONS

EXCEPTIONS contains information on violations of integrity constraints. This view is created by the utlexcpt.sql script.

Column	Datatype	NULL	Description
ROW_ID	ROWID		Row that caused the violation
OWNER	VARCHAR2(30)		Owner of the table
TABLE_NAME	VARCHAR2(30)		Name of the table
CONSTRAINT	VARCHAR2(30)		Integrity constraint that was violated

FLASHBACK_TRANSACTION_QUERY

FLASHBACK_TRANSACTION_QUERY displays information about all flashback transaction queries in the database.

Column	Datatype	NULL	Description
XID	RAW(8)		Transaction identifier
START_SCN	NUMBER		Transaction start system change number (SCN)
START_TIMESTAMP	DATE		Transaction start timestamp
COMMIT_SCN	NUMBER		Transaction commit system change number; NULL for active transactions
COMMIT_TIMESTAMP	DATE		Transaction commit timestamp; NULL for active transactions
LOGON_USER	VARCHAR2(30)		Logon user for the transaction
UNDO_CHANGE#	NUMBER		Undo system change number (1 or higher)
OPERATION	VARCHAR2(32)		Forward-going DML operation performed by the transaction: ■ D - Delete ■ I - Insert ■ U - Update ■ B ■ UNKNOWN
TABLE_NAME	VARCHAR2(256)		Name of the table to which the DML applies
TABLE_OWNER	VARCHAR2(32)		Owner of the table to which the DML applies
ROW_ID	VARCHAR2(19)		Rowid of the row that was modified by the DML
UNDO_SQL	VARCHAR2(4000)		SQL to undo the DML indicated by OPERATION

GLOBAL_NAME

GLOBAL_NAME contains one row that displays the global name of the current database.

Column	Datatype	NULL	Description
GLOBAL_NAME	VARCHAR2(4000)		Global name of the database

HS_ALL_CAPS

HS_ALL_CAPS contains information about all of the capabilities (that is, features) associated with non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER		Capability number
CONTEXT	NUMBER		Context in which this capability is applicable

Column	Datatype	NULL	Description
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Flag for internal use
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_ALL_DD

HS_ALL_DD contains data dictionary information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)		Data dictionary table name
TRANSLATION_TYPE	CHAR(1)		T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

HS_ALL_INITS

HS_ALL_INITS contains initialization parameter information about non-Oracle (FDS) data stores.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)		Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)		Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)		Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)		Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)		Name of the FDS instance

HS_BASE_CAPS

HS_BASE_CAPS contains information about base capability (that is, base features) of the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Description of the capability

HS_BASE_DD

HS_BASE_DD displays information from the base data dictionary translation table.

Column	Datatype	NULL	Description
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table

Column	Datatype	NULL	Description
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table

HS_CLASS_CAPS

HS_CLASS_CAPS contains information about the class-specific (driver) capabilities belonging to the non-Oracle (FDS) data store.

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Flag indicating the context in which the capability is enabled
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_CLASS_DD

HS_CLASS_DD displays information from the non-Oracle data store (FDS) class-specific data dictionary translations.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_CLASS_INIT

HS_CLASS_INIT displays information about the non-Oracle (FDS) class-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_CLASS_INIT_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

Column	Datatype	NULL	Description
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_EXTERNAL_OBJECT_PRIVILEGES

HS_EXTERNAL_OBJECT_PRIVILEGES contains information about the privileges on objects that are granted to users.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object; unique for each instance
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_EXTERNAL_OBJECTS

HS_EXTERNAL_OBJECTS contains information about all of the distributed external objects accessible from the Oracle server.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
OWNER	VARCHAR(30)	NOT NULL	Name of the user who created the object
OBJECT_NAME	VARCHAR2(30)	NOT NULL	Name of the object; unique for each instance
OBJECT_TYPE	VARCHAR2(13)	NOT NULL	Type of object: FUNCTION, PROCEDURE, PACKAGE, or LIBRARY
OBJECT_TEXT	LONG	NOT NULL	SQL text used to create the object

HS_EXTERNAL_USER_PRIVILEGES

HS_EXTERNAL_USER_PRIVILEGES contains information about all of the granted privileges that are not tied to any particular object.

Column	Datatype	NULL	Description
PRIVILEGE_NAME	VARCHAR2(30)	NOT NULL	Name of the privilege that was granted
GRANTEE	VARCHAR2(30)	NOT NULL	ID of the user that was granted the privilege
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS Class
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance

HS_FDS_CLASS

HS_FDS_CLASS contains information about legal non-Oracle (FDS) classes.

Column	Datatype	NULL	Description
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_CLASS_COMMENTS	VARCHAR2(255)		Text description of the non-Oracle class

Column	Datatype	NULL	Description
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_FDS_INST

HS_FDS_INST contains information about non-Oracle (FDS) instances.

Column	Datatype	NULL	Description
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_COMMENTS	VARCHAR2(255)		Text description of the non-Oracle instance
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_CAPS

HS_INST_CAPS contains information about instance-specific capabilities (that is, features).

Column	Datatype	NULL	Description
CAP_NUMBER	NUMBER	NOT NULL	Capability number
CAP_DESCRIPTION	VARCHAR2(255)		Capability description
CONTEXT	NUMBER		Context in which this capability is applicable
TRANSLATION	VARCHAR2(255)		Valid for functions; contains translation to FDS dialect
ADDITIONAL_INFO	NUMBER		Additional flags for internal use
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_DD

HS_INST_DD displays information from the non-Oracle (FDS) instance-specific data dictionary translations.

Column	Datatype	NULL	Description
DD_TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the Oracle data dictionary table
DD_TABLE_DESC	VARCHAR2(255)		Description of the Oracle data dictionary table
TRANSLATION_TYPE	CHAR(1)	NOT NULL	T = Translation, M = Mimic
TRANSLATION_TEXT	VARCHAR2(4000)		SQL statement containing the mapping
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example, ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
DD_TABLE_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

Column	Datatype	NULL	Description
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

HS_INST_INIT

HS_INST_INIT contains information about the non-Oracle (FDS) instance-specific initialization parameters.

Column	Datatype	NULL	Description
INIT_VALUE_NAME	VARCHAR2(64)	NOT NULL	Name of the initialization parameter
INIT_VALUE	VARCHAR2(255)	NOT NULL	Value of the initialization parameter
INIT_VALUE_TYPE	VARCHAR2(1)	NOT NULL	Environment variable (T or F). T means this is an environment variable; F means do not set as an environment variable
FDS_CLASS_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS class (for example: ODBC, DB2)
FDS_INST_NAME	VARCHAR2(30)	NOT NULL	Name of the FDS instance
FDS_INST_INIT_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_CLASS_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)
FDS_INST_ID	NUMBER	NOT NULL	Sequence: a counter that is incremented for every row inserted (used internally)

IDEPTREE

This view, created by utldtree.sql, lists the indented dependency tree. It is a pre-sorted, pretty-print version of DEPTREE.

Column	Datatype	NULL	Description
NESTED_LEVEL	NUMBER		Nesting level in the dependency tree
TYPE	VARCHAR2(15)		Object type
OWNER	VARCHAR2(30)		Object owner
NAME	VARCHAR2(1002)		Object name

IND

IND is a synonym for USER_INDEXES.

See Also: "USER_INDEXES" on page 5-112

INDEX_HISTOGRAM

INDEX_HISTOGRAM contains information from the ANALYZE INDEX ... VALIDATE STRUCTURE statement.

Note: The ANALYZE INDEX ... VALIDATE STRUCTURE OFFLINE statement must be used in order to collect statistics.

Column	Datatype	NULL	Description
REPEAT_COUNT	NUMBER		Number of times that one or more index keys is repeated in the table
KEYS_WITH_REPEAT_COUNT	NUMBER		Number of index keys that are repeated that many times

INDEX_STATS

INDEX_STATS stores information from the last ANALYZE INDEX ... VALIDATE STRUCTURE statement.

Note: The ANALYZE INDEX ... VALIDATE STRUCTURE OFFLINE statement must be used in order to collect statistics

Column	Datatype	NULL	Description
HEIGHT	NUMBER		Height of the B-Tree
BLOCKS	NUMBER	NOT NULL	Blocks allocated to the segment
NAME	VARCHAR2(30)	NOT NULL	Name of the index
PARTITION_NAME	VARCHAR2(30)		Name of the partition of the index which was analyzed. If the index is not partitioned, null is returned.
LF_ROWS	NUMBER		Number of leaf rows (values in the index)
LF_BLKS	NUMBER		Number of leaf blocks in the B-Tree
LF_ROWS_LEN	NUMBER		Sum of the lengths of all the leaf rows
LF_BLK_LEN	NUMBER		Usable space in a leaf block
BR_ROWS	NUMBER		Number of branch rows in the B-Tree
BR_BLKS	NUMBER		Number of branch blocks in the B-Tree
BR_ROWS_LEN	NUMBER		Sum of the lengths of all the branch blocks in the B-Tree
BR_BLK_LEN	NUMBER		Usable space in a branch block
DEL_LF_ROWS	NUMBER		Number of deleted leaf rows in the index
DEL_LF_ROWS_LEN	NUMBER		Total length of all deleted rows in the index
DISTINCT_KEYS	NUMBER		Number of distinct keys in the index (may include rows that have been deleted)
MOST_REPEATED_KEY	NUMBER		How many times the most repeated key is repeated (may include rows that have been deleted)
BTREE_SPACE	NUMBER		Total space currently allocated in the B-Tree
USED_SPACE	NUMBER		Total space that is currently being used in the B-Tree
PCT_USED	NUMBER		Percent of space allocated in the B-Tree that is being used
ROWS_PER_KEY	NUMBER		Average number of rows per distinct key (this figure is calculated without consideration of deleted rows)
BLKS_GETS_PER_ACCESS	NUMBER		Expected number of consistent mode block reads per row, assuming that a randomly chosen row is accessed using the index. Used to calculate the number of consistent reads that will occur during an index scan.
PRE_ROWS	NUMBER		Number of prefix rows (values in the index)
PRE_ROWS_LEN	NUMBER		Sum of lengths of all prefix rows
OPT_CMPR_COUNT	NUMBER		Optimal key compression length
OPT_CMPR_PCTSAVE	NUMBER		Corresponding space savings after an ANALYZE

LOGSTDBY_UNSUPPORTED_TABLES

LOGSTDBY_UNSUPPORTED_TABLES is a synonym for DBA_LOGSTDBY_UNSUPPORTED_TABLE.

See Also: "DBA_LOGSTDBY_UNSUPPORTED_TABLE" on page 4-149

MAP_OBJECT

MAP_OBJECT displays the hierarchical arrangement of storage containers for objects. Each row in the table represents a level in the hierarchy.

Column	Datatype	NULL	Description
OBJECT_NAME	VARCHAR2(2000)		Name of the object
OBJECT_OWNER	VARCHAR2(2000)		Owner of the object
OBJECT_TYPE	VARCHAR2(2000)		Object type
FILE_MAP_IDX	NUMBER		File index (corresponds to FILE_MAP_IDX in V$MAP_FILE)
DEPTH	NUMBER		Element depth within the I/O stack
ELEM_IDX	NUMBER		Index corresponding to the element
CU_SIZE	NUMBER		Contiguous set of logical blocks of the file (in HKB units) that is resident contiguously on the element
STRIDE	NUMBER		Number of HKB between contiguous units (CU) in the file that are contiguous on this element. Used in RAID5 and striped files.
NUM_CU	NUMBER		Number of contiguous units that are adjacent to each other on this element that are separated by STRIDE HKB in the file. In RAID5, the number of contiguous units also include the parity stripes.
ELEM_OFFSET	NUMBER		Element offset (in HKB units)
FILE_OFFSET	NUMBER		Offset (in HKB units) from the start of the file to the first byte of the contiguous units
DATA_TYPE	VARCHAR2(2000)		Datatype (DATA, PARITY, or DATA AND PARITY)
PARITY_POS	NUMBER		Position of the parity. Only for RAID5. This field is needed to distinguish the parity from the data part.
PARITY_PERIOD	NUMBER		Parity period. Only for RAID5.

NLS_DATABASE_PARAMETERS

NLS_DATABASE_PARAMETERS lists permanent NLS parameters of the database.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)	NOT NULL	Parameter name
VALUE	VARCHAR2(40)		Parameter value

NLS_INSTANCE_PARAMETERS

NLS_INSTANCE_PARAMETERS lists NLS parameters of the instance.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name

Column	Datatype	NULL	Description
VALUE	VARCHAR2(40)		Parameter value

NLS_SESSION_PARAMETERS

NLS_SESSION_PARAMETERS lists NLS parameters of the user session.

Column	Datatype	NULL	Description
PARAMETER	VARCHAR2(30)		Parameter name
VALUE	VARCHAR2(40)		Parameter value

OBJ

OBJ is a synonym for USER_OBJECTS.

See Also: "USER_OBJECTS" on page 5-119

PATH_VIEW

PATH_VIEW contains one row for each unique path to access a resource in the Oracle XML DB repository.

Column	Datatype	NULL	Description
PATH	VARCHAR2(1024)		An (absolute) path to repository resource RES
RES	XMLTYPE(XMLSchema "http://xmlns.oracle.com/xdb/XDBResource.xsd" Element "Resource")		The resource referred to by the PATH column
LINK	XMLTYPE		Link property
RESID	RAW(16)		Resource OID

See Also: *Oracle XML DB Developer's Guide* for information about using this view

PLAN_TABLE

PLAN_TABLE is automatically created as a global temporary table to hold the output of an EXPLAIN PLAN statement for all users. PLAN_TABLE is the default sample output table into which the EXPLAIN PLAN statement inserts rows describing execution plans.

While a PLAN_TABLE table is automatically set up for each user, you can use the SQL script utlxplan.sql to manually create a local PLAN_TABLE in your schema.

Column	Datatype	NULL	Description
STATEMENT_ID	VARCHAR2(30)		Value of the optional STATEMENT_ID parameter specified in the EXPLAIN PLAN statement
PLAN_ID	NUMBER		Unique identifier of a plan in the database
TIMESTAMP	DATE		Date and time when the EXPLAIN PLAN statement was generated

Column	Datatype	NULL	Description
REMARKS	VARCHAR2(4000)		Any comment (of up to 4000 bytes) you want to associate with each step of the explained plan. This column is used to indicate whether an outline or SQL Profile was used for the query. If you need to add or change a remark on any row of the PLAN_TABLE, then use the UPDATE statement to modify the rows of the PLAN_TABLE.
OPERATION	VARCHAR2(30)		Name of the internal operation performed in this step. In the first row generated for a statement, the column contains one of the following values: ▪ DELETE STATEMENT ▪ INSERT STATEMENT ▪ SELECT STATEMENT ▪ UPDATE STATEMENT
OPTIONS	VARCHAR2(255)		A variation on the operation described in the OPERATION column
OBJECT_NODE	VARCHAR2(128)		Name of the database link used to reference the object (a table name or view name). For local queries using parallel execution, this column describes the order in which output from operations is consumed.
OBJECT_OWNER	VARCHAR2(30)		Owner of the table or index
OBJECT_NAME	VARCHAR2(30)		Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)		Unique alias of a table or view in a SQL statement. For indexes, it is the object alias of the underlying table.
OBJECT_INSTANCE	NUMBER(38)		Number corresponding to the ordinal position of the object as it appears in the original statement. The numbering proceeds from left to right, outer to inner with respect to the original statement text. View expansion results in unpredictable numbers.
OBJECT_TYPE	VARCHAR2(30)		Modifier that provides descriptive information about the object; for example, NON-UNIQUE for indexes
OPTIMIZER	VARCHAR2(255)		Current mode of the optimizer
SEARCH_COLUMNS	NUMBER		Not currently used
ID	NUMBER(38)		A number assigned to each step in the execution plan
PARENT_ID	NUMBER(38)		ID of the next execution step that operates on the output of the ID step
DEPTH	NUMBER(38)		Depth of the operation in the row source tree that the plan represents. The value can be used for indenting the rows in a plan table report.
POSITION	NUMBER(38)		For the first row of output, this indicates the optimizer's estimated cost of executing the statement. For the other rows, it indicates the position relative to the other children of the same parent.
COST	NUMBER(38)		Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
CARDINALITY	NUMBER(38)		Estimate by the query optimization approach of the number of rows accessed by the operation
BYTES	NUMBER(38)		Estimate by the query optimization approach of the number of bytes accessed by the operation

Column	Datatype	NULL	Description
OTHER_TAG	VARCHAR2(255)		Describes the contents of the OTHER column: ■ SERIAL - Serial execution. Currently, SQL is not loaded in the OTHER column for this case. ■ SERIAL_FROM_REMOTE - Serial execution at a remote site. ■ PARALLEL_FROM_SERIAL - Serial execution. Output of step is partitioned or broadcast to parallel execution servers. ■ PARALLEL_TO_SERIAL - Parallel execution. Output of step is returned to serial query coordinator (QC) process. ■ PARALLEL_TO_PARALLEL - Parallel execution. Output of step is repartitioned to second set of parallel execution servers. ■ PARALLEL_COMBINED_WITH_PARENT - Parallel execution; Output of step goes to next step in same parallel process. No interprocess communication to parent. ■ PARALLEL_COMBINED_WITH_CHILD - Parallel execution. Input of step comes from prior step in same parallel process. No interprocess communication from child.
PARTITION_START	VARCHAR2(255)		Start partition of a range of accessed partitions: ■ *number* - Start partition has been identified by the SQL compiler, and its partition number is given by *number* ■ KEY - Start partition will be identified at run time from partitioning key values ■ ROW REMOVE_LOCATION - Start partition (same as the stop partition) will be computed at run time from the location of each record being retrieved. The record location is obtained by a user or from a global index. ■ INVALID - Range of accessed partitions is empty
PARTITION_STOP	VARCHAR2(255)		Stop partition of a range of accessed partitions: ■ *number* - Stop partition has been identified by the SQL compiler, and its partition number is given by *number* ■ KEY - Stop partition will be identified at run time from partitioning key values ■ ROW REMOVE_LOCATION - Stop partition (same as the start partition) will be computed at run time from the location of each record being retrieved. The record location is obtained by a user or from a global index. ■ INVALID - Range of accessed partitions is empty
PARTITION_ID	NUMBER(38)		Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	LONG		Other information that is specific to the execution step that a user might find useful (see the OTHER_TAG column)

Column	Datatype	NULL	Description
OTHER_XML	CLOB		Provides extra information specific to an execution step of the execution plan. The content of this column is structured using XML since multiple pieces of information can be stored there. This includes: ■ Name of the schema against which the query was parsed ■ Release number of the Oracle Database that produced the explain plan ■ Hash value associated with the execution plan ■ Name (if any) of the outline or the SQL profile used to build the execution plan ■ Indication of whether or not dynamic sampling was used to produce the plan ■ The outline data, a set of optimizer hints that can be used to regenerate the same plan
DISTRIBUTION	VARCHAR2(30)		Method used to distribute rows from producer query servers to consumer query servers **See Also:** *Oracle Database Data Warehousing Guide* for more information about consumer and producer query servers
CPU_COST	NUMBER(38)		CPU cost of the operation as estimated by the query optimizer's approach. The value of this column is proportional to the number of machine cycles required for the operation. For statements that use the rule-based approach, this column is NULL.
IO_COST	NUMBER(38)		I/O cost of the operation as estimated by the query optimizer's approach. The value of this column is proportional to the number of data blocks read by the operation. For statements that use the rule-based approach, this column is NULL.
TEMP_SPACE	NUMBER(38)		Temporary space (in bytes) used by the operation as estimated by the query optimizer's approach. For statements that use the rule-based approach, or for operations that do not use any temporary space, this column is NULL.
ACCESS_PREDICATES	VARCHAR2(4000)		Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)		Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)		Expressions produced by the operation
TIME	NUMBER(38)		Elapsed time (in seconds) of the operation as estimated by query optimization. For statements that use the rule-based approach, this column is NULL.
QBLOCK_NAME	VARCHAR2(30)		Name of the query block (either system-generated or defined by the user with the QB_NAME hint)

PLUGGABLE_SET_CHECK

PLUGGABLE_SET_CHECK contains pluggable set checks.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of the object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name
OBJ1_TYPE	VARCHAR2(15)		Object Type
TS1_NAME	VARCHAR2(30)		Tablespace containing Object 1

Column	Datatype	NULL	Description
OBJ2_NAME	VARCHAR2(30)		Object Name
OBJ2_SUBNAME	VARCHAR2(30)		SubObject2Name
OBJ2_TYPE	VARCHAR2(15)		Object Type
OBJ2_OWNER	VARCHAR2(30)		Object owner of second object
TS2_NAME	VARCHAR2(30)		Tablespace containing Object 1
CONSTRAINT_NAME	VARCHAR2(30)		Name of dependent constraint
REASON	VARCHAR2(79)		Reason for Pluggable check violation
MESG_ID	NUMBER		The message ID

PRODUCT_COMPONENT_VERSION

PRODUCT_COMPONENT_VERSION contains version and status information for component products.

Column	Datatype	NULL	Description
PRODUCT	VARCHAR2(64)		Product name
VERSION	VARCHAR2(64)		Version number
STATUS	VARCHAR2(64)		Status of release

PROXY_USERS

PROXY_USERS describes the list of proxy users and the clients on whose behalf they can act.

Column	Datatype	NULL	Description
PROXY	VARCHAR2(30)	NOT NULL	Name of a proxy user
CLIENT	VARCHAR2(30)	NOT NULL	Name of the client user who the proxy user can act as
AUTHENTICATION	VARCHAR2(3)		Indicates whether the proxy is required to supply the client's authentication credentials (YES) or not (NO)
FLAGS	VARCHAR2(35)		Flags associated with the proxy/client pair: ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ NO CLIENT ROLES MAY BE ACTIVATED ■ PROXY MAY ACTIVATE ROLE ■ PROXY MAY ACTIVATE ALL CLIENT ROLES ■ PROXY MAY NOT ACTIVATE ROLE

PSTUBTBL

This table contains information on stubs generated by the PSTUB utility so that an Oracle Forms 3.0 client can call stored procedures in the Oracle Database.

> **Note:** The contents of this table are intended only for use by the PSTUB utility.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Schema part of the identifier of a stored procedure

Column	Datatype	NULL	Description
DBNAME	VARCHAR2(128)		Database link part of the identifier of a stored procedure
LUN	VARCHAR2(30)		Library unit name part of the identifier of a stored procedure
LUTYPE	VARCHAR2(3)		Type of the stored procedure
LINENO	NUMBER		Line number of the stub
LINE	VARCHAR2(1800)		Text of the stub

PUBLIC_DEPENDENCY

PUBLIC_DEPENDENCY lists dependencies to and from objects, by object number.

Column	Datatype	NULL	Description
OBJECT_ID	NUMBER	NOT NULL	Object number
REFERENCED_OBJECT_ID	NUMBER	NOT NULL	Referenced object (the parent object)

PUBLICSYN

PUBLICSYN contains information on public synonyms.

Column	Datatype	NULL	Description
SNAME	VARCHAR2(30)		Name of the synonym
CREATOR	VARCHAR2(30)		Owner of the synonym
TNAME	VARCHAR2(30)		Table of which this is a synonym
DATABASE	VARCHAR2(128)		Database in which the table resides
TABTYPE	VARCHAR2(9)		Type of table

QUEUE_PRIVILEGES

QUEUE_PRIVILEGES shows all Advanced Queuing object privileges granted to the session.

Column	Datatype	NULL	Description
GRANTEE	VARCHAR2(30)	NOT NULL	Name of the user to whom access was granted
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
NAME	VARCHAR2(30)	NOT NULL	Name of the object
GRANTOR	VARCHAR2(30)	NOT NULL	Name of the user who performed the grant
ENQUEUE_PRIVILEGE	NUMBER		Permission to ENQUEUE to the queue
DEQUEUE_PRIVILEGE	NUMBER		Permission to DEQUEUE from the queue

RECYCLEBIN

RECYCLEBIN is a synonym for USER_RECYCLEBIN.

See Also: "USER_RECYCLEBIN" on page 5-123

REPORT_COMPONENTS

REPORT_COMPONENTS ???.

Column	Datatype	NULL	Description
COMPONENT_NAME	VARCHAR2(30)	NOT NULL	???
COMPONENT_DESCRIPTION	VARCHAR2(256)		???
REPORT_NAME	VARCHAR2(30)	NOT NULL	???
REPORT_DESCRIPTION	VARCHAR2(256)		???
SCHEMA_FILENAME	VARCHAR2(500)		???
SCHEMA_DATA	XMLTYPE		???

REPORT_FILES

REPORT_FILES ???.

Column	Datatype	NULL	Description
COMPONENT_NAME	VARCHAR2(30)		???
FILENAME	VARCHAR2(500)	NOT NULL	???
DATA	XMLTYPE		???

REPORT_FORMATS

REPORT_FORMATS ???.

Column	Datatype	NULL	Description
COMPONENT_NAME	VARCHAR2(30)	NOT NULL	???
REPORT_NAME	VARCHAR2(30)	NOT NULL	???
FORMAT_NAME	VARCHAR2(30)	NOT NULL	???
DESCRIPTION	VARCHAR2(256)		???
TYPE	VARCHAR2(6)		???
XSLT_FILENAME	VARCHAR2(500)		???
XSLT_DATA	XMLTYPE		???
TEXT_LINESIZE	NUMBER		???

RESOURCE_COST

RESOURCE_COST lists the cost for each resource.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the resource
UNIT_COST	NUMBER	NOT NULL	Cost of the resource

RESOURCE_MAP

RESOURCE_MAP describes resources. This table can be used to map resource names to resource numbers.

Column	Datatype	NULL	Description
RESOURCE#	NUMBER	NOT NULL	Numeric resource code
TYPE#	NUMBER	NOT NULL	Numeric type code

Column	Datatype	NULL	Description
NAME	VARCHAR2(32)	NOT NULL	Name of the resource

RESOURCE_VIEW

RESOURCE_VIEW contains one row for each resource in the Oracle XML DB repository.

Column	Datatype	NULL	Description
RES	XMLTYPE(XMLSchema "http://xmlns.oracle.com/xdb/XDBResource.xsd" Element "Resource")		A resource in the repository
ANY_PATH	VARCHAR2(4000)		An (absolute) path to the resource
RESID	RAW(16)		Resource OID, which is a unique handle to the resource

See Also: *Oracle XML DB Developer's Guide* for information about using this view

ROLE_ROLE_PRIVS

ROLE_ROLE_PRIVS describes the roles granted to other roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
GRANTED_ROLE	VARCHAR2(30)	NOT NULL	Role that was granted
ADMIN_OPTION	VARCHAR2(3)		Signifies that the role was granted with ADMIN option

ROLE_SYS_PRIVS

ROLE_SYS_PRIVS describes system privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege granted to the role
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN option (YES) or not (NO)

ROLE_TAB_PRIVS

ROLE_TAB_PRIVS describes table privileges granted to roles. Information is provided only about roles to which the user has access.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role
OWNER	VARCHAR2(30)	NOT NULL	Owner of the object
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the object

Column	Datatype	NULL	Description
COLUMN_NAME	VARCHAR2(30)		Name of the column, if applicable
PRIVILEGE	VARCHAR2(40)	NOT NULL	Object privilege granted to the role
GRANTABLE	VARCHAR2(3)		YES if the role was granted with ADMIN OPTION; otherwise NO

SCHEDULER_BATCH_ERRORS

SCHEDULER_BATCH_ERRORS displays the errors caused by each call in the batch after a Scheduler batch call (when the COMMIT_SEMANTICS argument has been set to ABSORB_ERRORS).

Column	Datatype	NULL	Description
ARRAY_INDEX	NUMBER		Index of the job in the batch
OBJECT_TYPE	VARCHAR2(30)		Object type: ■ JOB ■ LIGHTWEIGHT JOB ■ UNKNOWN
OBJECT_NAME	VARCHAR2(100)		Full name of the object (including schema)
ATTR_NAME	VARCHAR2(30)		Name of the attribute being set (if this is a batch set attribute call); NULL otherwise
ERROR_CODE	NUMBER		Top level error code
ERROR_MESSAGE	VARCHAR2(4000)		Complete error stack
ADDITIONAL_INFO	VARCHAR2(4000)		Additional information (currently unused)

SEQ

SEQ is a synonym for USER_SEQUENCES.

See Also: "USER_SEQUENCES" on page 5-127

SESSION_CONTEXT

SESSION_CONTEXT describes the context attributes and their values set for the current session.

Column	Datatype	NULL	Description
NAMESPACE	VARCHAR2(30)		Namespace that the active attribute is in
ATTRIBUTE	VARCHAR2(30)		Name of the active attribute
VALUE	VARCHAR2(4000)		Value of the active attribute

See Also: *Oracle Database SQL Language Reference* for information on the predefined attributes of the current session

SESSION_PRIVS

SESSION_PRIVS describes the privileges that are currently available to the user.

Column	Datatype	NULL	Description
PRIVILEGE	VARCHAR2(40)	NOT NULL	Name of the privilege

SESSION_ROLES

SESSION_ROLES describes the roles that are currently enabled to the user.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(30)	NOT NULL	Name of the role

SOURCE_SIZE

Oracle accesses this view to create views about object size.

See Also: "DBA_OBJECT_SIZE" on page 5-3 and "USER_OBJECT_SIZE" on page 5-119

STMT_AUDIT_OPTION_MAP

STMT_AUDIT_OPTION_MAP describes auditing option type codes. This table can be used to map auditing option type numbers to type names.

Column	Datatype	NULL	Description
OPTION#	NUMBER	NOT NULL	Numeric auditing option type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of auditing option
PROPERTY	NUMBER	NOT NULL	Property flag of the auditing option

SYN

SYN is a synonym for USER_SYNONYMS.

See Also: "ALL_SYNONYMS" on page 3-64

SYNONYMS

SYNONYMS is included for compatibility. Oracle recommends that you do not use this view.

SYS_OBJECTS

SYS_OBJECTS maps object IDs to object types and segment data block addresses.

Column	Datatype	NULL	Description
OBJECT_TYPE	VARCHAR2(15)		Type of the object
OBJECT_TYPE_ID	NUMBER		Type ID of the object
SEGMENT_TYPE_ID	NUMBER		Type of segment: TABLE, CLUSTER, INDEX, ROLLBACK, DEFERRED ROLLBACK, TEMPORARY, CACHE
OBJECT_ID	NUMBER		Object identifier
HEADER_FILE	NUMBER		ID of the file containing the segment header
HEADER_BLOCK	NUMBER		ID of the block containing the segment header
TS_NUMBER	NUMBER		The tablespace number

SYSCATALOG

SYSCATALOG is included for compatibility. Oracle recommends that you do not use this view.

SYSFILES

SYSFILES is included for compatibility. Oracle recommends that you do not use this view.

SYSSEGOBJ

SYSSEGOBJ is included for compatibility. Oracle recommends that you do not use this view.

SYSTEM_PRIVILEGE_MAP

SYSTEM_PRIVILEGE_MAP describes privilege type codes. This table can be used to map privilege type numbers to type names.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege
PROPERTY	NUMBER	NOT NULL	Property flag of the privilege

TAB

TAB is included for compatibility. Oracle recommends that you do not use this view.

TABLE_PRIVILEGE_MAP

TABLE_PRIVILEGE_MAP describes privilege (auditing option) type codes. This table can be used to map privilege (auditing option) type numbers to type names.

Column	Datatype	NULL	Description
PRIVILEGE	NUMBER	NOT NULL	Numeric privilege (auditing option) type code
NAME	VARCHAR2(40)	NOT NULL	Name of the type of privilege (auditing option)

TABQUOTAS

TABQUOTAS is included for compatibility. Oracle recommends that you do not use this view.

TABS

TABS is a synonym for USER_TABLES.

See Also: "USER_TABLES" on page 5-134

TRUSTED_SERVERS

TRUSTED_SERVERS displays whether a server is trusted or untrusted.

Column	Datatype	NULL	Description
TRUST	VARCHAR2(9)		Trustedness of the server listed. Values can be TRUSTED or UNTRUSTED servers which are not listed in the NAME column have opposite trustedness.
NAME	VARCHAR2(128)		Server name. Can be a specific server name or ALL for all servers.

Table 5–1 shows examples of the values returned depending on the status of the servers.

Table 5–1 TRUSTED_SERVERS Values

Condition (If . . .)	TRUSTED column	NAME column
... all servers are trusted	Trusted	ALL
... no servers are trusted	Untrusted	ALL
... all servers except DB1 are trusted	Untrusted	DB1
... all servers except DB1 are untrusted	Trusted	DB1

See Also: *Oracle Database Heterogeneous Connectivity Administrator's Guide*

TS_PITR_CHECK

This view, created by `catpitr.sql`, provides information on any dependencies or restrictions that might prevent tablespace point-in-time recovery from proceeding. This view applies only to the tablespace point-in-time recovery feature.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)	NOT NULL	The owner of the object preventing tablespace point-in-time recovery. See the REASON column for details.
OBJ1_NAME	VARCHAR2(30)	NOT NULL	The name of the object preventing tablespace point-in-time recovery
OBJ1_TYPE	VARCHAR2(15)		The object type for the object preventing tablespace point-in-time recovery
OBJ1_SUBNAME	VARCHAR2(30)		Subordinate to OBJ1_NAME
TS1_NAME	VARCHAR2(30)	NOT NULL	Name of the tablespace containing the object preventing tablespace point-in-time recovery
OBJ2_NAME	VARCHAR2(30)		The name of a second object which may be preventing tablespace point-in-time recovery. If NULL, object 1 is the only object preventing recovery.
OBJ2_TYPE	VARCHAR2(15)		The object type for the second object (will be NULL if OBJ2_NAME is NULL)
OBJ2_OWNER	VARCHAR2(30)		The owner of the second object (will be NULL if OBJ2_NAME is NULL)
OBJ2_SUBNAME	VARCHAR2(30)		Subordinate to OBJ2_NAME
TS2_NAME	VARCHAR2(30)		Name of the tablespace containing second object which may be preventing tablespace point-in-time recovery (-1 indicates not applicable)
CONSTRAINT_NAME	VARCHAR2(30)		Name of the constraint
REASON	VARCHAR2(78)		Reason why tablespace point-in-time recovery cannot proceed

See Also: *Oracle Database Backup and Recovery Basics*

TS_PITR_OBJECTS_TO_BE_DROPPED

TS_PITR_OBJECTS_TO_BE_DROPPED lists all objects lost as a result of performing tablespace point-in-time recovery. This view applies only to the tablespace point-in-time recovery feature.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)	NOT NULL	The owner of the object
NAME	VARCHAR2(30)	NOT NULL	The name of the object that will be lost as a result of undergoing tablespace point-in-time recovery
CREATION_TIME	DATE	NOT NULL	Creation timestamp of the object
TABLESPACE_NAME	VARCHAR2(30)		Name of the tablespace containing the object

UNI_PLUGGABLE_SET_CHECK

UNI_PLUGGABLE_SET_CHECK contains pluggable check information.

Column	Datatype	NULL	Description
OBJ1_OWNER	VARCHAR2(30)		Owner of object
OBJ1_NAME	VARCHAR2(30)		Object 1
OBJ1_SUBNAME	VARCHAR2(30)		SubObject1Name
OBJ1_TYPE	VARCHAR2(15)		Object Type
TS1_NAME	VARCHAR2(30)		Tablespace containing Object 1
OBJ2_NAME	VARCHAR2(30)		Object Name
OBJ2_SUBNAME	VARCHAR2(30)		SubObject2Name
OBJ2_TYPE	VARCHAR2(15)		Object Type
OBJ2_OWNER	VARCHAR2(30)		Object owner of second object
TS2_NAME	VARCHAR2(30)		Tablespace containing Object 1
CONSTRAINT_NAME	VARCHAR2(30)		Name of dependent constraint
REASON	VARCHAR2(79)		Reason for Pluggable check violation
MESG_ID	NUMBER		The message ID

USER_ADDM_FDG_BREAKDOWN

USER_ADDM_FDG_BREAKDOWN describes the contribution for each finding from the different instances owned by the current user. Its columns are the same as those in "DBA_ADDM_FDG_BREAKDOWN" on page 4-2.

USER_ADDM_FINDINGS

USER_ADDM_FINDINGS displays the ADDM findings discovered by the advisors owned by the current user. Each row for ADDM tasks in the related USER_ADVISOR_FINDINGS view has a corresponding row in this view. Its columns (except for OWNER) are the same as those in "DBA_ADDM_FINDINGS" on page 4-2.

USER_ADDM_INSTANCES

USER_ADDM_INSTANCES provides instance-level information for ADDM tasks that finished executing in all instances owned by the current user. Its columns are the same as those in "DBA_ADDM_INSTANCES" on page 4-3.

USER_ADDM_TASKS

USER_ADDM_TASKS displays information about the ADDM tasks owned by the current user. The view contains one row for each row in the related USER_ADVISOR_TASKS view that has ADVISOR_NAME=ADDM and STATUS=COMPLETED. Its columns (except for OWNER) are the same as those in "DBA_ADVISOR_TASKS" on page 4-27.

USER_ADVISOR_ACTIONS

USER_ADVISOR_ACTIONS displays information about the actions associated with the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_ACTIONS.

See Also: "DBA_ADVISOR_ACTIONS" on page 4-6

USER_ADVISOR_DIR_TASK_INST

USER_ADVISOR_DIR_TASK_INST ???.

USER_ADVISOR_EXECUTIONS

USER_ADVISOR_EXECUTIONS displays metadata information for tasks owned by the current user. Its columns (except for OWNER) are the same as those in"DBA_ADVISOR_EXECUTIONS" on page 4-10.

USER_ADVISOR_EXEC_PARAMETERS

USER_ADVISOR_EXEC_PARAMETERS describes task parameters that are specific to a given task execution owned by the current user. Its columns (except for OWNER) are the same as those in "DBA_ADVISOR_EXEC_PARAMETERS" on page 4-11.

USER_ADVISOR_FDG_BREAKDOWN

USER_ADVISOR_FDG_BREAKDOWN ???.

USER_ADVISOR_FINDINGS

USER_ADVISOR_FINDINGS displays the findings discovered by the advisors owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_FINDINGS.

See Also: "DBA_ADVISOR_FINDINGS" on page 4-12

USER_ADVISOR_JOURNAL

USER_ADVISOR_JOURNAL displays the journal entries for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_JOURNAL.

See Also: "DBA_ADVISOR_JOURNAL" on page 4-13

USER_ADVISOR_LOG

USER_ADVISOR_LOG displays information about the current state of the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_LOG.

See Also: "DBA_ADVISOR_LOG" on page 4-13

USER_ADVISOR_OBJECTS

USER_ADVISOR_OBJECTS displays information about the objects currently referenced by the advisors owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_OBJECTS.

See Also: "DBA_ADVISOR_OBJECTS" on page 4-15

USER_ADVISOR_PARAMETERS

USER_ADVISOR_PARAMETERS displays the task parameters and their current values for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_PARAMETERS.

See Also: "DBA_ADVISOR_PARAMETERS" on page 4-16

USER_ADVISOR_RATIONALE

USER_ADVISOR_RATIONALE displays information about the rationales for the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_RATIONALE.

See Also: "DBA_ADVISOR_RATIONALE" on page 4-17

USER_ADVISOR_RECOMMENDATIONS

USER_ADVISOR_RECOMMENDATIONS displays the results of an analysis of the recommendations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_RECOMMENDATIONS.

See Also: "DBA_ADVISOR_RECOMMENDATIONS" on page 4-18

USER_ADVISOR_SQLA_REC_SUM

USER_ADVISOR_SQLA_REC_SUM displays recommendation rollup information for the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_REC_SUM.

See Also: "DBA_ADVISOR_SQLA_REC_SUM" on page 4-19

USER_ADVISOR_SQLA_TABLES

USER_ADVISOR_SQLA_TABLES ???.

USER_ADVISOR_SQLA_WK_MAP

USER_ADVISOR_SQLA_WK_MAP displays the workload references for the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_WK_MAP.

See Also: "DBA_ADVISOR_SQLA_WK_MAP" on page 4-20

USER_ADVISOR_SQLA_WK_STMTS

USER_ADVISOR_SQLA_WK_STMTS displays information about the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLA_WK_STMTS.

See Also: "DBA_ADVISOR_SQLA_WK_STMTS" on page 4-20

USER_ADVISOR_SQLPLANS

USER_ADVISOR_SQLPLANS displays a list of all SQL execution plans owned by the current user. Its columns are the same as those in "DBA_ADVISOR_SQLPLANS" on page 4-21.

USER_ADVISOR_SQLSTATS

USER_ADVISOR_SQLSTATS displays a list of all SQL compilation and execution statistics owned by the current user. Its columns are the same as those in "DBA_ADVISOR_SQLSTATS" on page 4-23.

USER_ADVISOR_SQLW_JOURNAL

USER_ADVISOR_SQLW_JOURNAL displays the journal entries for the workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_JOURNAL.

See Also: "DBA_ADVISOR_SQLW_JOURNAL" on page 4-23

USER_ADVISOR_SQLW_PARAMETERS

USER_ADVISOR_SQLW_PARAMETERS displays the workload parameters and their current values owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_PARAMETERS.

See Also: "DBA_ADVISOR_SQLW_PARAMETERS" on page 4-24

USER_ADVISOR_SQLW_STMTS

USER_ADVISOR_SQLW_STMTS displays rows that correspond to the statements in the workload owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_STMTS.

See Also: "DBA_ADVISOR_SQLW_STMTS" on page 4-25

USER_ADVISOR_SQLW_SUM

USER_ADVISOR_SQLW_SUM displays an aggregated picture of the SQLWkld workload objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_SUM.

See Also: "DBA_ADVISOR_SQLW_SUM" on page 4-26

USER_ADVISOR_SQLW_TABLES

USER_ADVISOR_SQLW_TABLES displays cross references between the workload statements and the tables referenced in the statement. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_TABLES.

See Also: "DBA_ADVISOR_SQLW_TABLES" on page 4-27

USER_ADVISOR_SQLW_TEMPLATES

USER_ADVISOR_SQLW_TEMPLATES displays an aggregated picture of the SQLWkld template objects owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_SQLW_TEMPLATES.

See Also: "DBA_ADVISOR_SQLW_TEMPLATES" on page 4-27

USER_ADVISOR_TASKS

USER_ADVISOR_TASKS displays information about the tasks owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_TASKS.

See Also: "DBA_ADVISOR_TASKS" on page 4-27

USER_ADVISOR_TEMPLATES

USER_ADVISOR_TEMPLATES displays information about the templates owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ADVISOR_TEMPLATES.

See Also: "DBA_ADVISOR_TEMPLATES" on page 4-29

USER_ALL_TABLES

USER_ALL_TABLES describes the object tables and relational tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ALL_TABLES.

See Also: "ALL_ALL_TABLES" on page 2-5

USER_AQ_AGENT_PRIVS

USER_AQ_AGENT_PRIVS displays information about the registered AQ agents that are mapped to the current user. Its columns (except for DB_USERNAME) are the same as those in DBA_AQ_AGENT_PRIVS.

See Also: "DBA_AQ_AGENT_PRIVS" on page 4-34

USER_ARGUMENTS

USER_ARGUMENTS lists the arguments of the procedures and functions that are owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ARGUMENTS.

See Also: "ALL_ARGUMENTS" on page 2-13

USER_ASSEMBLIES

USER_ASSEMBLIES provides information about all assemblies owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ASSEMBLIES.

See Also: "ALL_ASSEMBLIES" on page 2-15

USER_ASSOCIATIONS

USER_ASSOCIATIONS describes user-defined statistics associated with objects owned by the current user. Its columns are the same as those in ALL_ASSOCIATIONS.

See Also: "ALL_ASSOCIATIONS" on page 2-16

USER_ATTRIBUTE_TRANSFORMATIONS

USER_ATTRIBUTE_TRANSFORMATIONS displays information about the transformation functions for the transformations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_ATTRIBUTE_TRANSFORMATIONS.

See Also: "DBA_ATTRIBUTE_TRANSFORMATIONS" on page 4-35

USER_AUDIT_OBJECT

This view, created by cataudit.sql, lists audit trail records for statements concerning objects that are accessible to the current user. Its columns are the same as those in "DBA_AUDIT_OBJECT" on page 4-37.

USER_AUDIT_POLICIES

USER_AUDIT_POLICIES describes the fine-grained auditing policies on the tables and views owned by the current user. Its columns (except for OBJECT_SCHEMA) are the same as those in ALL_AUDIT_POLICIES.

See Also: "ALL_AUDIT_POLICIES" on page 2-16

USER_AUDIT_POLICY_COLUMNS

USER_AUDIT_POLICY_COLUMNS describes the fine-grained auditing policy columns on the tables and views owned by the current user. Its columns (except for OBJECT_SCHEMA) are the same as those in ALL_AUDIT_POLICY_COLUMNS.

See Also: "ALL_AUDIT_POLICY_COLUMNS" on page 2-17

USER_AUDIT_SESSION

This view, created by cataudit.sql, lists all audit trail records concerning connections and disconnections for the user. Its columns are the same as those in "DBA_AUDIT_SESSION" on page 4-38.

USER_AUDIT_STATEMENT

This view, created by cataudit.sql, lists audit trail entries for the following statements issued by the user: GRANT, REVOKE, AUDIT, NOAUDIT, and ALTER SYSTEM. Its columns are the same as those in "DBA_AUDIT_STATEMENT" on page 4-39.

USER_AUDIT_TRAIL

This view, created by cataudit.sql, lists audit trail entries relevant to the user. Its columns are the same as those in "DBA_AUDIT_TRAIL" on page 4-41.

USER_AW_PS

USER_AW_PS describes the page spaces in the analytic workspaces owned by the current user. Its columns (except for OWNER) are the same as those in ALL_AW_PS.

See Also: "ALL_AW_PS" on page 2-18

USER_AWS

USER_AWS describes the analytic workspaces owned by the current user. Its columns (except for OWNER) are the same as those in ALL_AWS.

See Also: "ALL_AWS" on page 2-18

USER_BASE_TABLE_MVIEWS

USER_BASE_TABLE_MVIEWS describes the materialized views using materialized view logs owned by the current user. Its columns are the same as those in ALL_BASE_TABLE_MVIEWS.

See Also: "ALL_BASE_TABLE_MVIEWS" on page 2-19

USER_CATALOG

USER_CATALOG lists tables, views, clusters, synonyms, and sequences owned by the current user. Its columns are the same as those in "ALL_CATALOG" on page 2-24.

USER_CHANGE_NOTIFICATION_REGS

USER_CHANGE_NOTIFICATION_REGS describes the change notification registrations owned by the current user. Its columns (except for USERNAME) are the same as those in DBA_CHANGE_NOTIFICATION_REGS.

See Also: "DBA_CHANGE_NOTIFICATION_REGS" on page 4-52

USER_CLU_COLUMNS

USER_CLU_COLUMNS maps columns in the current user's tables to cluster columns. Its columns are the same as those in "DBA_CLU_COLUMNS" on page 4-52.

USER_CLUSTER_HASH_EXPRESSIONS

USER_CLUSTER_HASH_EXPRESSIONS lists hash functions for the hash clusters owned by the current user. Its columns are the same as those in "ALL_CLUSTER_HASH_EXPRESSIONS" on page 2-27.

USER_CLUSTERS

USER_CLUSTERS describes all the clusters owned by the current user. Its columns are the same as those in "ALL_CLUSTERS" on page 2-27.

USER_COL_COMMENTS

USER_COL_COMMENTS displays comments on the columns of the tables and views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_COL_COMMENTS.

See Also: "ALL_COL_COMMENTS" on page 2-28

USER_COL_PENDING_STATS

USER_COL_PENDING_STATS describes the pending statistics of the columns owned by the current user. Its columns (except for OWNER) are the same as those in ALL_COL_PENDING_STATS.

See Also: "ALL_COL_PENDING_STATS" on page 2-29

USER_COL_PRIVS

USER_COL_PRIVS describes the column object grants for which the current user is the object owner, grantor, or grantee. Its columns are the same as those in DBA_COL_PRIVS.

See Also: "DBA_COL_PRIVS" on page 4-53

USER_COL_PRIVS_MADE

USER_COL_PRIVS_MADE describes the column object grants for which the current user is the object owner. Its columns (except for OWNER) are the same as those in ALL_COL_PRIVS_MADE.

See Also: "ALL_COL_PRIVS_MADE" on page 2-30

USER_COL_PRIVS_RECD

USER_COL_PRIVS_RECD describes the column object grants for which the current user is the grantee. Its columns (except for GRANTEE) are the same as those in ALL_COL_PRIVS_RECD.

See Also: "ALL_COL_PRIVS_RECD" on page 2-30

USER_COLL_TYPES

`USER_COLL_TYPES` describes named collection types (VARRAYs, nested tables, object tables, and so on) in the current user's schema. Its columns are the same as those in "ALL_COLL_TYPES" on page 2-31.

USER_COMPARISON

`USER_COMPARISON` displays details about comparison objects owned by the current user. Its columns (except for OWNER) are the same as those in "DBA_COMPARISON" on page 4-55.

USER_COMPARISON_COLUMNS

`USER_COMPARISON_COLUMNS` displays details about the columns for all comparison objects owned by the current user. Its columns (except for OWNER) are the same as those in "DBA_COMPARISON_COLUMNS" on page 4-56.

USER_COMPARISON_ROW_DIF

`USER_COMPARISON_ROW_DIF` provides details about differing rows in all comparison scans owned by the current user. Its columns are the same as those in "DBA_COMPARISON_ROW_DIF" on page 4-56.

USER_COMPARISON_SCAN

`USER_COMPARISON_SCAN` displays details about all comparison scans owned by the current user. Its columns (except for OWNER) are the same as those in "DBA_COMPARISON_SCAN" on page 4-57.

USER_COMPARISON_SCAN_VALUES

`USER_COMPARISON_SCAN_VALUES` displays details about the values for all comparison scans owned by the current user. Its columns (except for `OWNER`) are the same as those in "DBA_COMPARISON_SCAN_VALUES" on page 4-57.

USER_CONS_COLUMNS

`USER_CONS_COLUMNS` describes columns that are owned by the current user and that are specified in constraint definitions. Its columns are the same as those in "ALL_CONS_COLUMNS" on page 2-31.

USER_CONS_OBJ_COLUMNS

`USER_CONS_OBJ_COLUMNS` displays information about the types that object columns (or attributes) or collection elements have been constrained to, in the tables owned by the current user. Its columns (except for `OWNER`) are the same as those in `ALL_CONS_OBJ_COLUMNS`.

See Also: "ALL_CONS_OBJ_COLUMNS" on page 2-32

USER_CONSTRAINTS

USER_CONSTRAINTS describes all constraint definitions on tables owned by the current user. Its columns are the same as those in "ALL_CONSTRAINTS" on page 2-32.

USER_CQ_NOTIFICATION_QUERIES

USER_CQ_NOTIFICATION_QUERIES describes the registered queries for the CQ notifications owned by the current user. Its columns (except for USERNAME) are the same as those in DBA_CQ_NOTIFICATION_QUERIES.

See Also: "DBA_CQ_NOTIFICATION_QUERIES" on page 4-59

USER_CUBE_ATTR_VISIBILITY

USER_CUBE_ATTR_VISIBILITY describes the OLAP attributes visible for the dimensions, hierarchies, and levels owned by the current user Its columns (except for OWNER) are the same as those in ALL_CUBE_ATTR_VISIBILITY.

USER_CUBE_ATTRIBUTES

USER_CUBE_ATTRIBUTES describes the attributes for the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_ATTRIBUTES.

USER_CUBE_BUILD_PROCESSES

USER_CUBE_BUILD_PROCESSES describes the OLAP build processes and maintenance scripts owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_BUILD_PROCESSES.

USER_CUBE_CALCULATED_MEMBERS

USER_CUBE_CALCULATED_MEMBERS describes the calculated members for the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_CALCULATED_MEMBERS.

USER_CUBE_DIM_LEVELS

USER_CUBE_DIM_LEVELS describes the OLAP dimension levels owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIM_LEVELS.

USER_CUBE_DIM_MODELS

USER_CUBE_DIM_MODELS describes the models for the OLAP dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIM_MODELS.

USER_CUBE_DIM_VIEW_COLUMNS

USER_CUBE_DIM_VIEW_COLUMNS describes the columns of the relational views of the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIM_VIEW_COLUMNS.

USER_CUBE_DIM_VIEWS

USER_CUBE_DIM_VIEWS describes the relational views of the OLAP dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIM_VIEWS.

USER_CUBE_DIMENSIONALITY

USER_CUBE_DIMENSIONALITY describes the dimension order for the OLAP cubes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIMENSIONALITY.

USER_CUBE_DIMENSIONS

USER_CUBE_DIMENSIONS describes the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_DIMENSIONS.

USER_CUBE_HIER_LEVELS

USER_CUBE_HIER_LEVELS describes the hierarchy levels for the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_HIER_LEVELS.

USER_CUBE_HIER_VIEW_COLUMNS

USER_CUBE_HIER_VIEW_COLUMNS describes the columns of the relational hierarchy views of the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_HIER_VIEW_COLUMNS.

USER_CUBE_HIER_VIEWS

USER_CUBE_HIER_VIEWS describes the hierarchies for the OLAP cube dimensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_HIER_VIEWS.

USER_CUBE_HIERARCHIES

USER_CUBE_HIERARCHIES describes the OLAP dimension hierarchies owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_HIERARCHIES.

USER_CUBE_MEASURES

USER_CUBE_MEASURES describes the measures for the OLAP cubes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_MEASURES.

USER_CUBE_VIEW_COLUMNS

USER_CUBE_VIEW_COLUMNS describes the columns of relational views of OLAP cubes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_VIEW_COLUMNS.

USER_CUBE_VIEWS

USER_CUBE_VIEWS describes the relational views of the OLAP cubes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBE_VIEWS.

USER_CUBES

USER_CUBES describes the OLAP cubes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_CUBES.

USER_DATAPUMP_JOBS

USER_DATAPUMP_JOBS displays the Data Pump jobs owned by the current user. Its columns (except for OWNER_NAME) are the same as those in DBA_DATAPUMP_JOBS.

See Also: "DBA_DATAPUMP_JOBS" on page 4-63

USER_DB_LINKS

USER_DB_LINKS describes the database links owned by the current user. Its columns are the same as those in ALL_DB_LINKS except that it does not have the OWNER column. It also displays an additional column, PASSWORD, which is no longer used and for which nothing is returned. The PASSWORD column is maintained for backward compatibility only.

See Also: "ALL_DB_LINKS" on page 2-44

USER_DEPENDENCIES

USER_DEPENDENCIES describes dependencies between procedures, packages, functions, package bodies, and triggers owned by the current user, including dependencies on views created without any database links. Its columns are the same as those in "ALL_DEPENDENCIES" on page 2-46.

USER_DIM_ATTRIBUTES

USER_DIM_ATTRIBUTES describes the relationship between dimension levels and functionally dependent columns in the current user's schema. The level columns and the dependent column must be in the same table. This view's columns are the same as those in "ALL_DIM_ATTRIBUTES" on page 2-47.

USER_DIM_CHILD_OF

USER_DIM_CHILD_OF describes a hierarchical relationship of 1 to *n* between pairs of levels in dimensions owned by the current user. Its columns are the same as those in "ALL_DIM_CHILD_OF" on page 2-47.

USER_DIM_HIERARCHIES

USER_DIM_HIERARCHIES describes the dimension hierarchies owned by the current user. Its columns are the same as those in "ALL_DIM_HIERARCHIES" on page 2-48.

USER_DIM_JOIN_KEY

USER_DIM_JOIN_KEY describes the join between two dimension tables owned by the current user. The join is always specified between a parent dimension level column and a child column. This view's columns are the same as those in "ALL_DIM_JOIN_KEY" on page 2-48.

USER_DIM_LEVEL_KEY

USER_DIM_LEVEL_KEY describes columns of dimension levels owned by the current user. This view's columns are the same as those in "ALL_DIM_LEVEL_KEY" on page 2-49.

USER_DIM_LEVELS

USER_DIM_LEVELS describes the levels of dimensions owned by the current user. All columns of a dimension level must come from the same relation. This view's columns are the same as those in "ALL_DIM_LEVELS" on page 2-49.

USER_DIMENSIONS

USER_DIMENSIONS describes dimension objects in the user's schema. Its columns are the same as those in "ALL_DIMENSIONS" on page 2-49.

USER_ENCRYPTED_COLUMNS

USER_ENCRYPTED_COLUMNS maintains encryption algorithm information for all encrypted columns in all tables in the user's schema. Its columns (except for OWNER) are the same as those in "ALL_ENCRYPTED_COLUMNS" on page 2-50.

USER_EPG_DAD_AUTHORIZATION

USER_EPG_DAD_AUTHORIZATION describes the DADs that are authorized to use the user's privileges. Its columns (except for USERNAME) are the same as those in "DBA_EPG_DAD_AUTHORIZATION" on page 4-67.

USER_ERRORS

USER_ERRORS describes the current errors on the stored objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_ERRORS.

See Also: "ALL_ERRORS" on page 2-51

USER_EVALUATION_CONTEXT_TABLES

USER_EVALUATION_CONTEXT_TABLES describes the tables in the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in ALL_EVALUATION_CONTEXT_TABLES.

See Also: "ALL_EVALUATION_CONTEXT_TABLES" on page 2-51

USER_EVALUATION_CONTEXT_VARS

USER_EVALUATION_CONTEXT_VARS describes the variables in the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in ALL_EVALUATION_CONTEXT_VARS.

See Also: "ALL_EVALUATION_CONTEXT_VARS" on page 2-52

USER_EVALUATION_CONTEXTS

USER_EVALUATION_CONTEXTS describes the rule evaluation contexts owned by the current user. Its columns (except for EVALUATION_CONTEXT_OWNER) are the same as those in ALL_EVALUATION_CONTEXTS.

See Also: "ALL_EVALUATION_CONTEXTS" on page 2-52

USER_EXTENTS

USER_EXTENTS describes the extents comprising the segments owned by the current user's objects. Its columns (except for OWNER, FILE_ID, BLOCK_ID, and RELATIVE_FNO) are the same as those in DBA_EXTENTS.

See Also: "DBA_EXTENTS" on page 4-69

USER_EXTERNAL_LOCATIONS

USER_EXTERNAL_LOCATIONS describes the locations (data sources) of the external tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_EXTERNAL_LOCATIONS.

See Also: "ALL_EXTERNAL_LOCATIONS" on page 2-53

USER_EXTERNAL_TABLES

USER_EXTERNAL_TABLES describes the external tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_EXTERNAL_TABLES.

See Also: "ALL_EXTERNAL_TABLES" on page 2-53

USER_FILE_GROUP_EXPORT_INFO

USER_FILE_GROUP_EXPORT_INFO shows export-related information for all file groups that the current user can manage. Its columns (except for FILE_GROUP_OWNER) are the same as those in "ALL_FILE_GROUP_EXPORT_INFO" on page 2-54.

USER_FILE_GROUP_FILES

USER_FILE_GROUP_FILES shows the file set for each versioned group owned by the current user. Its columns (except for FILE_GROUP_OWNER) are the same as those in"ALL_FILE_GROUP_FILES" on page 2-55.

USER_FILE_GROUP_TABLES

USER_FILE_GROUP_TABLES shows information about tables owned by the current user that can be imported using the file set. Its columns (except for FILE_GROUP_OWNER) are the same as those in "ALL_FILE_GROUP_TABLES" on page 2-55.

USER_FILE_GROUP_TABLESPACES

USER_FILE_GROUP_TABLESPACES shows information about the transportable tablespaces present (partially or completely) in the file set owned by the current user (when the file set contains dump files). Its columns (except for FILE_GROUP_OWNER) are the same as those in "ALL_FILE_GROUP_TABLESPACES" on page 2-56.

USER_FILE_GROUP_VERSIONS

USER_FILE_GROUP_VERSIONS shows top-level version information for all file groups owned by the current user. Its columns (except for FILE_GROUP_OWNER) are the same as those in "ALL_FILE_GROUP_VERSIONS" on page 2-56.

USER_FILE_GROUPS

USER_FILE_GROUPS shows top-level metadata about file groups owned by the current user. Its columns (except for FILE_GROUP_OWNER) are the same as those in "ALL_FILE_GROUPS" on page 2-57.

USER_FLASHBACK_ARCHIVE

USER_FLASHBACK_ARCHIVE describes the flashback archives available to the current user. Its columns are the same as those in DBA_FLASHBACK_ARCHIVE.

See Also: "DBA_FLASHBACK_ARCHIVE" on page 4-75

USER_FLASHBACK_ARCHIVE_TABLES

USER_FLASHBACK_ARCHIVE_TABLES displays information about the tables owned by the current user that are enabled for Flashback Archive. Its columns are the same as those in DBA_FLASHBACK_ARCHIVE_TABLES.

See Also: "DBA_FLASHBACK_ARCHIVE_TABLES" on page 4-76

USER_FLASHBACK_TXN_REPORT

USER_FLASHBACK_TXN_REPORT displays information about the compensating transactions owned by the current user that have been committed in the database. Its columns (except for USER#) are the same as those in DBA_FLASHBACK_TXN_REPORT.

See Also: "DBA_FLASHBACK_TXN_REPORT" on page 4-76

USER_FLASHBACK_TXN_STATE

USER_FLASHBACK_TXN_STATE displays information about the compensating status of the transactions owned by the current user. Its columns (except for USER#) are the same as those in DBA_FLASHBACK_TXN_STATE.

See Also: "DBA_FLASHBACK_TXN_STATE" on page 4-77

USER_FREE_SPACE

`USER_FREE_SPACE` describes the free extents in the tablespaces accessible to the current user. Its columns are the same as those in `DBA_FREE_SPACE`.

See Also: "DBA_FREE_SPACE" on page 4-77

USER_IDENTIFIERS

`USER_IDENTIFIERS` displays information about the identifiers in the stored objects owned by the current user. Its columns (except for `OWNER`) are the same as those in `ALL_IDENTIFIERS`.

See Also: "ALL_IDENTIFIERS" on page 2-57

USER_IND_COLUMNS

`USER_IND_COLUMNS` describes the columns of the indexes owned by the current user and columns of indexes on tables owned by the current user. Its columns are the same as those in "ALL_IND_COLUMNS" on page 2-58.

USER_IND_EXPRESSIONS

`USER_IND_EXPRESSIONS` describes expressions of function-based indexes on tables owned by the current user. Its columns are the same as those in "ALL_IND_EXPRESSIONS" on page 2-59.

USER_IND_PARTITIONS

`USER_IND_PARTITIONS` describes, for each index partition owned by the current user, the partition-level partitioning information, the storage parameters for the partition, and various partition statistics analyzed by `ANALYZE` statements. Its columns are the same as those in "ALL_IND_PARTITIONS" on page 2-59.

USER_IND_PENDING_STATS

`USER_IND_PENDING_STATS` describes pending statistics for all tables, partitions, and subpartitions owned by the current user. Its columns (except for `OWNER`) are the same as those in "ALL_IND_PENDING_STATS" on page 2-60.

USER_IND_STATISTICS

`USER_IND_STATISTICS` displays optimizer statistics for the indexes on the tables owned by the current user. Its columns (except for `OWNER`) are the same as those in `ALL_IND_STATS`.

See Also: "ALL_IND_STATISTICS" on page 2-61

USER_IND_SUBPARTITIONS

`USER_IND_SUBPARTITIONS` describes, for each index subpartition owned by the current user, the partition-level partitioning information, the storage parameters for the subpartition, and various partition statistics collected by ANALYZE statements. Its columns are the same as those in "ALL_IND_SUBPARTITIONS" on page 2-62.

USER_INDEXES

USER_INDEXES describes indexes owned by the current user. To gather statistics for this view, use the SQL ANALYZE statement. This view supports parallel partitioned index scans. Its columns (except for OWNER) are the same as those in "ALL_INDEXES" on page 2-64.

USER_INDEXTYPE_ARRAYTYPES

USER_INDEXTYPE_ARRAYTYPES displays information about the array types specified by the indextypes owned by the current user. Its columns are the same as those in ALL_INDEXTYPE_ARRAYTYPES.

See Also: "ALL_INDEXTYPE_ARRAYTYPES" on page 2-66

USER_INDEXTYPE_COMMENTS

USER_INDEXTYPE_COMMENTS displays comments for the user-defined indextypes owned by the current user. Its columns are the same as those in ALL_INDEXTYPE_COMMENTS.

See Also: "ALL_INDEXTYPE_COMMENTS" on page 2-67

USER_INDEXTYPE_OPERATORS

USER_INDEXTYPE_OPERATORS lists all the operators supported by indextypes owned by the current user. Its columns are the same as those in ALL_INDEXTYPE_OPERATORS.

See Also: "ALL_INDEXTYPE_OPERATORS" on page 2-67

USER_INDEXTYPES

USER_INDEXTYPES describes the indextypes owned by the current user. Its columns are the same as those in ALL_INDEXTYPES.

See Also: "ALL_INDEXTYPES" on page 2-68

USER_INTERNAL_TRIGGERS

USER_INTERNAL_TRIGGERS describes the internal triggers on all tables owned by the current user. Its columns are the same as those in "ALL_INTERNAL_TRIGGERS" on page 2-69.

USER_JAVA_ARGUMENTS

USER_JAVA_ARGUMENTS displays argument information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_ARGUMENTS.

See Also: "ALL_JAVA_ARGUMENTS" on page 2-69

USER_JAVA_CLASSES

USER_JAVA_CLASSES displays class level information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_CLASSES.

See Also: "ALL_JAVA_CLASSES" on page 2-70

USER_JAVA_COMPILER_OPTIONS

USER_JAVA_COMPILER_OPTIONS displays information about the native compiler options owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_COMPILER_OPTIONS.

See Also: "ALL_JAVA_COMPILER_OPTIONS" on page 2-70

USER_JAVA_DERIVATIONS

USER_JAVA_DERIVATIONS displays mapping information about Java source objects and their derived Java class objects and Java resource objects for the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_DERIVATIONS.

See Also: "ALL_JAVA_DERIVATIONS" on page 2-71

USER_JAVA_FIELDS

USER_JAVA_FIELDS displays field information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_FIELDS.

See Also: "ALL_JAVA_FIELDS" on page 2-71

USER_JAVA_IMPLEMENTS

USER_JAVA_IMPLEMENTS describes interfaces implemented by the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_IMPLEMENTS.

See Also: "ALL_JAVA_IMPLEMENTS" on page 2-72

USER_JAVA_INNERS

USER_JAVA_INNERS displays information about inner classes referred to by the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_INNERS.

See Also: "ALL_JAVA_INNERS" on page 2-73

USER_JAVA_LAYOUTS

USER_JAVA_LAYOUTS displays class layout information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_LAYOUTS.

See Also: "ALL_JAVA_LAYOUTS" on page 2-74

USER_JAVA_METHODS

USER_JAVA_METHODS displays method information about the stored Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_METHODS.

See Also: "ALL_JAVA_METHODS" on page 2-74

USER_JAVA_NCOMPS

USER_JAVA_NCOMPS displays ncomp-related information about the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_NCOMPS.

See Also: "ALL_JAVA_NCOMPS" on page 2-75

USER_JAVA_POLICY

USER_JAVA_POLICY describes Java security permissions for the current user. Its columns are the same as those in DBA_JAVA_POLICY.

See Also: "DBA_JAVA_POLICY" on page 4-138

USER_JAVA_RESOLVERS

USER_JAVA_RESOLVERS displays information about resolvers of the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_RESOLVERS.

See Also: "ALL_JAVA_RESOLVERS" on page 2-76

USER_JAVA_THROWS

USER_JAVA_THROWS displays information about exceptions thrown from methods of the Java classes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_JAVA_THROWS.

See Also: "ALL_JAVA_THROWS" on page 2-76

USER_JOBS

USER_JOBS describes the jobs owned by the current user. Its columns are the same as those in DBA_JOBS.

See Also: "DBA_JOBS" on page 4-139

USER_JOIN_IND_COLUMNS

USER_JOIN_IND_COLUMNS describes all join conditions owned by the current user. Its columns are the same as those in "ALL_JOIN_IND_COLUMNS" on page 2-77.

USER_LIBRARIES

USER_LIBRARIES describes the libraries owned by the current user. Its columns (except for OWNER) are the same as those in ALL_LIBRARIES.

See Also: "ALL_LIBRARIES" on page 2-77

USER_LOB_PARTITIONS

USER_LOB_PARTITIONS displays the LOB partitions contained in tables owned by the current user. Its columns are the same as those in "ALL_LOB_PARTITIONS" on page 2-78.

USER_LOB_SUBPARTITIONS

USER_LOB_SUBPARTITIONS describes partition-level attributes of LOB data subpartitions owned by the current user. Its columns are the same as those in "ALL_LOB_SUBPARTITIONS" on page 2-79.

- DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.

USER_LOB_TEMPLATES

USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. Its columns (except for USER_NAME) are the same as those in ALL_LOB_TEMPLATES.

See Also: "ALL_LOB_TEMPLATES" on page 2-80

USER_LOBS

USER_LOBS displays the user's CLOBs and BLOBs contained in the user's tables. BFILEs are stored outside the database, so they are not described by this view. This view's columns are the same as those in "ALL_LOBS" on page 2-81.

USER_LOG_GROUP_COLUMNS

USER_LOG_GROUP_COLUMNS describes columns that are owned by the current user and that are specified in log groups. Its columns are the same as those in "ALL_LOG_GROUP_COLUMNS" on page 2-81.

USER_LOG_GROUPS

USER_LOG_GROUPS describes log group definitions on tables owned by the current user. Its columns are the same as those in "ALL_LOG_GROUPS" on page 2-82.

USER_MEASURE_FOLDER_CONTENTS

USER_MEASURE_FOLDER_CONTENTS describes the contents of the OLAP measure folders owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MEASURE_FOLDER_CONTENTS.

USER_MEASURE_FOLDERS

USER_MEASURE_FOLDERS describes the OLAP measure folders owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MEASURE_FOLDERS.

USER_METHOD_PARAMS

USER_METHOD_PARAMS describes the method parameters of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_METHOD_PARAMS.

See Also: "ALL_METHOD_PARAMS" on page 2-83

USER_METHOD_RESULTS

USER_METHOD_RESULTS describes the method results of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_METHOD_RESULTS.

See Also: "ALL_METHOD_RESULTS" on page 2-84

USER_MINING_MODEL_ATTRIBUTES

USER_MINING_MODEL_ATTRIBUTES describes the mining model attributes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MINING_MODEL_ATTRIBUTES.

See Also: "ALL_MINING_MODEL_ATTRIBUTES" on page 2-85

USER_MINING_MODEL_SETTINGS

USER_MINING_MODEL_SETTINGS describes the mining model settings owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MINING_MODEL_SETTINGS.

See Also: "ALL_MINING_MODEL_SETTINGS" on page 2-86

USER_MINING_MODELS

USER_MINING_MODELS describes the mining models owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MINING_MODELS.

See Also: "ALL_MINING_MODELS" on page 2-86

USER_MVIEW_AGGREGATES

USER_MVIEW_AGGREGATES describes the grouping functions (aggregated measures) that appear in the SELECT list of aggregated materialized views owned by the current user. Its columns are the same as those in "ALL_MVIEW_AGGREGATES" on page 2-87.

USER_MVIEW_ANALYSIS

USER_MVIEW_ANALYSIS describes all materialized views owned by the current user that potentially support query rewrite and that provide additional information for analysis by applications. Its columns are the same as those in "ALL_MVIEW_ANALYSIS" on page 2-88.

Note: This view excludes materialized views that reference remote tables or that include references to non-static values such as SYSDATE or USER. This view also excludes materialized views that were created as snapshots prior to Oracle8*i* and that were never altered to enable query rewrite.

USER_MVIEW_COMMENTS

USER_MVIEW_COMMENTS displays comments on the materialized views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_MVIEW_COMMENTS.

See Also: "ALL_MVIEW_COMMENTS" on page 2-89

USER_MVIEW_DETAIL_PARTITION

USER_MVIEW_DETAIL_PARTITION displays freshness information for all materialized views, with respect to a PCT detail partition, owned by the current user. Its columns are the same as those in "ALL_MVIEW_DETAIL_PARTITION" on page 2-90.

USER_MVIEW_DETAIL_RELATIONS

USER_MVIEW_DETAIL_RELATIONS represents the named detail relations that are either in the FROM list of a materialized view, or that are indirectly referenced through views in the FROM list. Its columns are the same as those in "ALL_MVIEW_DETAIL_RELATIONS" on page 2-90.

USER_MVIEW_DETAIL_SUBPARTITION

USER_MVIEW_DETAIL_SUBPARTITION displays freshness information for all materialized views, with respect to a PCT detail subpartition, owned by the current user. Its columns are the same as those in "ALL_MVIEW_DETAIL_SUBPARTITION" on page 2-91.

USER_MVIEW_JOINS

USER_MVIEW_JOINS describes a join between two columns in the WHERE clause of a subquery that defines a materialized view. Its columns are the same as those in "ALL_MVIEW_JOINS" on page 2-91.

USER_MVIEW_KEYS

USER_MVIEW_KEYS describes the columns or expressions in the SELECT list upon which materialized views in the current user's schema are based. Its columns are the same as those in "ALL_MVIEW_KEYS" on page 2-92.

USER_MVIEW_LOGS

USER_MVIEW_LOGS describes all materialized view logs owned by the current user. Its columns are the same as those in ALL_MVIEW_LOGS.

See Also: "ALL_MVIEW_LOGS" on page 2-92

USER_MVIEW_REFRESH_TIMES

USER_MVIEW_REFRESH_TIMES describes refresh times of the materialized views owned by the current user. Its columns are the same as those in ALL_MVIEW_REFRESH_TIMES.

See Also: "ALL_MVIEW_REFRESH_TIMES" on page 2-93

USER_MVIEWS

USER_MVIEWS describes all materialized views owned by the current user. Its columns are the same as those in ALL_MVIEWS.

See Also: "ALL_MVIEWS" on page 2-94

USER_NESTED_TABLE_COLS

USER_NESTED_TABLE_COLS describes the columns of the nested tables owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_NESTED_TABLE_COLS" on page 2-97. To gather statistics for this view, use the ANALYZE SQL statement.

USER_NESTED_TABLES

USER_NESTED_TABLES describes the nested tables in tables owned by the current user. Its columns are the same as those in "ALL_NESTED_TABLES" on page 2-99.

USER_NETWORK_ACL_PRIVILEGES

USER_NETWORK_ACL_PRIVILEGES describes the status of the network privileges for the current user to access network hosts.

Column	Datatype	NULL	Description
HOST	VARCHAR2(1000)		Network host
LOWER_PORT	NUMBER(5)		Lower bound of the port range
UPPER_PORT	NUMBER(5)		Upper bound of the port range
PRIVILEGE	CHAR(7)		Network privilege
STATUS	VARCHAR2(7)		Privilege status: ▪ DENIED ▪ GRANTED

USER_OBJ_AUDIT_OPTS

This view, created by cataudit.sql, lists auditing characteristics for all objects owned by the current user. Its columns are the same as those in "DBA_OBJ_AUDIT_OPTS" on page 5-2.

USER_OBJ_COLATTRS

USER_OBJ_COLATTRS describes object columns and attributes contained in the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_OBJ_COLATTRS.

See Also: "ALL_OBJ_COLATTRS" on page 2-99

USER_OBJECT_SIZE

USER_OBJECT_SIZE lists the sizes, in bytes, of various PL/SQL objects. Its columns are the same as those in "DBA_OBJECT_SIZE" on page 5-3.

USER_OBJECT_TABLES

USER_OBJECT_TABLES describes the object tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_OBJECT_TABLES.

See Also: "ALL_OBJECT_TABLES" on page 2-100

USER_OBJECTS

USER_OBJECTS describes all objects owned by the current user. Its columns are the same as those in "ALL_OBJECTS" on page 2-102.

USER_OPANCILLARY

USER_OPANCILLARY provides ancillary information for operators owned by the current user. Its columns are the same as those in "ALL_OPANCILLARY" on page 2-103.

USER_OPARGUMENTS

USER_OPARGUMENTS provides argument information for operator bindings owned by the current user. Its columns are the same as those in "ALL_OPARGUMENTS" on page 2-103.

USER_OPBINDINGS

USER_OPBINDINGS describes the binding functions and methods on the operators owned by the current user. Its columns are the same as those in ALL_OPBINDINGS.

See Also: "ALL_OPBINDINGS" on page 2-103

USER_OPERATOR_COMMENTS

USER_OPERATOR_COMMENTS displays comments for the user-defined operators owned by the current user. Its columns are the same as those in ALL_OPERATOR_COMMENTS.

See Also: "ALL_OPERATOR_COMMENTS" on page 2-104

USER_OPERATORS

USER_OPERATORS describes all operators owned by the current user. Its columns are the same as those in ALL_OPERATORS.

See Also: "ALL_OPERATORS" on page 2-105

USER_OUTLINE_HINTS

USER_OUTLINE_HINTS describes the set of hints stored in the outlines owned by the current user. Its columns (except for OWNER) are the same as those in DBA_OUTLINE_HINTS.

See Also: "DBA_OUTLINE_HINTS" on page 5-5

USER_OUTLINES

USER_OUTLINES describes the stored outlines owned by the current user. Its columns (except for OWNER) are the same as those in DBA_OUTLINES.

See Also: "DBA_OUTLINES" on page 5-5

USER_PART_COL_STATISTICS

USER_PART_COL_STATISTICS displays column statistics and histogram information for the table partitions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_COL_STATISTICS.

See Also: "ALL_PART_COL_STATISTICS" on page 3-1

USER_PART_HISTOGRAMS

USER_PART_HISTOGRAMS displays the histogram data (endpoints per histogram) for the histograms on the table partitions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_HISTOGRAMS.

See Also: "ALL_PART_HISTOGRAMS" on page 3-2

USER_PART_INDEXES

USER_PART_INDEXES displays the object-level partitioning information for the partitioned indexes owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_INDEXES.

See Also: "ALL_PART_INDEXES" on page 3-2

USER_PART_KEY_COLUMNS

USER_PART_KEY_COLUMNS describes the partitioning key columns for the partitioned objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_KEY_COLUMNS.

See Also: "ALL_PART_KEY_COLUMNS" on page 3-4

USER_PART_LOBS

USER_PART_LOBS displays table-level information about the partitioned LOBs owned by the current user, including default attributes for LOB data partitions. Its columns (except for TABLE_OWNER) are the same as those in ALL_PART_LOBS.

See Also: "ALL_PART_LOBS" on page 3-5

USER_PART_TABLES

USER_PART_TABLES displays the object-level partitioning information for the partitioned tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PART_TABLES.

See Also: "ALL_PART_TABLES" on page 3-7

USER_PARTIAL_DROP_TABS

USER_PARTIAL_DROP_TABS describes all tables in the schema of the current user that have partially completed DROP COLUMN operations. Its columns are the same as those in "ALL_PARTIAL_DROP_TABS" on page 3-9.

USER_PASSWORD_LIMITS

USER_PASSWORD_LIMITS describes the password profile parameters that are assigned to the user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the password resource
LIMIT	VARCHAR2(40)		Value of the resource limit

USER_PENDING_CONV_TABLES

USER_PENDING_CONV_TABLES describes the pending conversion tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PENDING_CONV_TABLES.

See Also: "ALL_PENDING_CONV_TABLES" on page 3-9

USER_PLSQL_OBJECT_SETTINGS

USER_PLSQL_OBJECT_SETTINGS displays information about the compiler settings for the stored objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_PLSQL_OBJECT_SETTINGS.

See Also: "ALL_PLSQL_OBJECT_SETTINGS" on page 3-10

USER_POLICIES

USER_POLICIES describes all Oracle Virtual Private Database (VPD) security policies associated with objects owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICIES.

See Also: "ALL_POLICIES" on page 3-10

USER_POLICY_CONTEXTS

USER_POLICY_CONTEXTS describes the driving contexts defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICY_CONTEXTS.

See Also: "ALL_POLICY_CONTEXTS" on page 3-12

USER_POLICY_GROUPS

USER_POLICY_GROUPS describes the policy groups defined for the synonyms, tables, and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_POLICY_GROUPS.

See Also: "ALL_POLICY_GROUPS" on page 3-12

USER_PROCEDURES

USER_PROCEDURES lists all functions and procedures, along with their associated properties. Its columns (except OWNER) are the same as those in "ALL_PROCEDURES" on page 3-12.

USER_PROXIES

USER_PROXIES displays information about connections the current user is allowed to proxy. Its columns (except for PROXY) are the same as those in DBA_PROXIES.

See Also: "DBA_PROXIES" on page 5-9

USER_PUBLISHED_COLUMNS

USER_PUBLISHED_COLUMNS describes the source table columns owned by the current user which have been published for the source tables. This view is intended for use by Change Data Capture subscribers. Its columns (except CHANGE_TABLE_SCHEMA and CHANGE_TABLE_NAME) are the same as those in DBA_PUBLISHED_COLUMNS.

See Also: "DBA_PUBLISHED_COLUMNS" on page 5-10

USER_QUEUE_SCHEDULES

USER_QUEUE_SCHEDULES describes the propagation schedules whose source queues are owned by the current user. Its columns (except for SCHEMA) are the same as those in DBA_QUEUE_SCHEDULES.

See Also: "DBA_QUEUE_SCHEDULES" on page 5-11

USER_QUEUE_SUBSCRIBERS

USER_QUEUE_SUBSCRIBERS returns the list of subscribers on queues that are under the current user's schema. Its columns are the same as those in "ALL_QUEUE_SUBSCRIBERS" on page 3-14.

USER_QUEUE_TABLES

USER_QUEUE_TABLES describes the queues in the queue tables created in the current user's schema. Its columns (except for OWNER) are the same as those in ALL_QUEUE_TABLES.

See Also: "ALL_QUEUE_TABLES" on page 3-15

USER_QUEUES

USER_QUEUES describes the operational characteristics of every queue in the user's schema. Its columns are the same as those in "ALL_QUEUES" on page 3-16.

See Also: *Oracle Streams Advanced Queuing User's Guide* for more information about these views and Advanced Queuing

USER_RECYCLEBIN

USER_RECYCLEBIN displays information about the recycle bin owned by the current user. Its columns (except for OWNER) are the same as those in DBA_RECYCLEBIN.

See Also: "DBA_RECYCLEBIN" on page 5-14

USER_REFRESH

USER_REFRESH describes all refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH" on page 3-16.

USER_REFRESH_CHILDREN

USER_REFRESH_CHILDREN lists all the objects in refresh groups owned by the current user. Its columns are the same as those in "ALL_REFRESH_CHILDREN" on page 3-17.

USER_REFS

USER_REFS describes the REF columns and REF attributes in the object type columns of tables owned by the current user. Its columns are the same as those in "ALL_REFS" on page 3-18.

USER_REGISTERED_MVIEWS

USER_REGISTERED_MVIEWS describes all registered materialized views (registered at a master site or a master materialized view site) owned by the current user. Its columns are the same as those in ALL_REGISTERED_MVIEWS.

See Also: "ALL_REGISTERED_MVIEWS" on page 3-19

USER_REGISTRY

USER_REGISTRY displays information about the components loaded into the database that are owned by the current user. Its columns are the same as those in DBA_REGISTRY.

See Also: "DBA_REGISTRY" on page 5-18

USER_RESOURCE_LIMITS

USER_RESOURCE_LIMITS displays the resource limits for the current user.

Column	Datatype	NULL	Description
RESOURCE_NAME	VARCHAR2(32)	NOT NULL	Name of the resource
LIMIT	VARCHAR2(40)		Limit placed on this resource

USER_RESUMABLE

USER_RESUMABLE lists resumable statements executed by the current user. Its columns are the same as those in DBA_RESUMABLE.

See Also: "DBA_RESUMABLE" on page 5-20

USER_REWRITE_EQUIVALENCES

USER_REWRITE_EQUIVALENCES describes the rewrite equivalences owned by the current user. Its columns are the same as those in ALL_REWRITE_EQUIVALENCES.

See Also: "ALL_REWRITE_EQUIVALENCES" on page 3-19

USER_ROLE_PRIVS

USER_ROLE_PRIVS describes the roles granted to the current user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name of the user, or PUBLIC
GRANTED_ROLE	VARCHAR2(30)		Name of the role granted to the user
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN OPTION (YES) or not (NO)
DEFAULT_ROLE	VARCHAR2(3)		Indicates whether the role is designated as a DEFAULT ROLE for the user (YES) or not (NO)
OS_GRANTED	VARCHAR2(3)		Granted by the operating system (YES) or not (NO); occurs if configuration parameter OS_ROLES = true.

USER_RSRC_CONSUMER_GROUP_PRIVS

USER_RSRC_CONSUMER_GROUP_PRIVS displays information about the resource consumer groups to which the current user is assigned. Its columns (except for GRANTEE) are the same as those in DBA_RSRC_CONSUMER_GROUP_PRIVS.

See Also: "DBA_RSRC_CONSUMER_GROUP_PRIVS" on page 5-23

USER_RSRC_MANAGER_SYSTEM_PRIVS

USER_RSRC_MANAGER_SYSTEM_PRIVS displays information about the users who are granted system privileges for the DBMS_RESOURCE_MANAGER package. Its columns (except for GRANTEE) are the same as those in DBA_RSRC_MANAGER_SYSTEM_PRIVS.

See Also: "DBA_RSRC_MANAGER_SYSTEM_PRIVS" on page 5-25

USER_RULE_SET_RULES

USER_RULE_SET_RULES describes the rules in the rule sets owned by the current user. Its columns (except for RULE_SET_OWNER) are the same as those in ALL_RULE_SET_RULES.

See Also: "ALL_RULE_SET_RULES" on page 3-20

USER_RULE_SETS

USER_RULE_SETS describes the rule sets owned by the current user. Its columns (except for RULE_SET_OWNER) are the same as those in ALL_RULE_SETS.

See Also: "ALL_RULE_SETS" on page 3-21

USER_RULES

USER_RULES describes the rules owned by the current user. Its columns (except for RULE_OWNER) are the same as those in ALL_RULES.

See Also: "ALL_RULES" on page 3-21

USER_SCHEDULER_CHAIN_RULES

USER_SCHEDULER_CHAIN_RULES displays information about the rules for the chains owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_CHAIN_RULES.

See Also: "ALL_SCHEDULER_CHAIN_RULES" on page 3-22

USER_SCHEDULER_CHAIN_STEPS

USER_SCHEDULER_CHAIN_STEPS displays information about the defined steps of the chains owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_CHAIN_STEPS.

See Also: "ALL_SCHEDULER_CHAIN_STEPS" on page 3-22

USER_SCHEDULER_CHAINS

USER_SCHEDULER_CHAINS displays information about the chains owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_CHAINS.

See Also: "ALL_SCHEDULER_CHAINS" on page 3-23

USER_SCHEDULER_CREDENTIALS

USER_SCHEDULER_CREDENTIALS displays information about the credentials owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_CREDENTIALS.

See Also: "ALL_SCHEDULER_CREDENTIALS" on page 3-24

USER_SCHEDULER_JOB_ARGS

USER_SCHEDULER_JOB_ARGS displays information about the arguments of the Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_JOB_ARGS.

See Also: "ALL_SCHEDULER_JOB_ARGS" on page 3-25

USER_SCHEDULER_JOB_LOG

USER_SCHEDULER_JOB_LOG displays log information for the Scheduler jobs owned by the current user. Its columns are the same as those in ALL_SCHEDULER_JOB_LOG.

See Also: "ALL_SCHEDULER_JOB_LOG" on page 3-26

USER_SCHEDULER_JOB_RUN_DETAILS

USER_SCHEDULER_JOB_RUN_DETAILS displays log run details for the Scheduler jobs owned by the current user. Its columns are the same as those in ALL_SCHEDULER_JOB_RUN_DETAILS.

See Also: "ALL_SCHEDULER_JOB_RUN_DETAILS" on page 3-27

USER_SCHEDULER_JOBS

USER_SCHEDULER_JOBS displays information about the Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_JOBS.

See Also: "ALL_SCHEDULER_JOBS" on page 3-28

USER_SCHEDULER_PROGRAM_ARGS

USER_SCHEDULER_PROGRAM_ARGS displays information about the arguments of the Scheduler programs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_PROGRAM_ARGS.

See Also: "ALL_SCHEDULER_PROGRAM_ARGS" on page 3-31

USER_SCHEDULER_PROGRAMS

USER_SCHEDULER_PROGRAMS displays information about the Scheduler programs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_PROGRAMS.

See Also: "ALL_SCHEDULER_PROGRAMS" on page 3-32

USER_SCHEDULER_REMOTE_JOBSTATE

USER_SCHEDULER_REMOTE_JOBSTATE displays information about the state of the jobs owned by the current user at remote databases. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_REMOTE_JOBSTATE.

See Also: "ALL_SCHEDULER_REMOTE_JOBSTATE" on page 3-33

USER_SCHEDULER_RUNNING_CHAINS

USER_SCHEDULER_RUNNING_CHAINS displays information about the chain steps of the running chains owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_RUNNING_CHAINS.

See Also: "ALL_SCHEDULER_RUNNING_CHAINS" on page 3-34

USER_SCHEDULER_RUNNING_JOBS

USER_SCHEDULER_RUNNING_JOBS displays information about the running Scheduler jobs owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_RUNNING_JOBS.

See Also: "ALL_SCHEDULER_RUNNING_JOBS" on page 3-35

USER_SCHEDULER_SCHEDULES

USER_SCHEDULER_SCHEDULES displays information about the Scheduler schedules owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SCHEDULER_SCHEDULES.

See Also: "ALL_SCHEDULER_SCHEDULES" on page 3-36

USER_SEC_RELEVANT_COLS

USER_SEC_RELEVANT_COLS describes the security relevant columns of the security policies for the tables and views owned by the current user. Its columns (except for OBJECT_OWNER) are the same as those in ALL_SEC_RELEVANT_COLS.

See Also: "ALL_SEC_RELEVANT_COLS" on page 3-40

USER_SECONDARY_OBJECTS

USER_SECONDARY_OBJECTS provides information about secondary objects associated with domain indexes owned by the current user. This view is only relevant in the context of domain indexes. Its columns are the same as those in "ALL_SECONDARY_OBJECTS" on page 3-40.

USER_SEGMENTS

USER_SEGMENTS describes the storage allocated for the segments owned by the current user's objects. Its columns (except for OWNER, HEADER_FILE, HEADER_BLOCK, and RELATIVE_FNO) are the same as those in DBA_SEGMENTS.

See Also: "DBA_SEGMENTS" on page 5-32

USER_SEQUENCES

USER_SEQUENCES describes all sequences owned by the current user. Its columns are the same as those in "ALL_SEQUENCES" on page 3-41.

USER_SOURCE

USER_SOURCE describes the text source of the stored objects owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_SOURCE" on page 3-42.

USER_SOURCE_TABLES

USER_SOURCE_TABLES describes the source tables owned by the current user for which a change table is defined. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in DBA_SOURCE_TABLES.

See Also: "DBA_SOURCE_TABLES" on page 5-35

USER_SQLJ_TYPE_ATTRS

USER_SQLJ_TYPE_ATTRS describes the attributes of the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPE_ATTRS.

See Also: "ALL_SQLJ_TYPE_ATTRS" on page 3-42

USER_SQLJ_TYPE_METHODS

USER_SQLJ_TYPE_METHODS describes the methods of the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPE_METHODS.

See Also: "ALL_SQLJ_TYPE_METHODS" on page 3-43

USER_SQLJ_TYPES

USER_SQLJ_TYPES describes the SQLJ object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_SQLJ_TYPES.

See Also: "ALL_SQLJ_TYPES" on page 3-44

USER_SQLSET

USER_SQLSET displays information about the SQL tuning sets owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_SQLSET" on page 3-45.

USER_SQLSET_BINDS

USER_SQLSET_BINDS displays the bind values associated with the SQL tuning sets owned by the current user. Its columns (except for SQLSET_OWNER) are the same as those in "ALL_SQLSET_BINDS" on page 3-45.

USER_SQLSET_PLANS

USER_SQLSET_PLANS describes captured plans for statements in the SQL tuning sets owned by the current user. Its columns (except for SQLSET_OWNER) are the same as those in "ALL_SQLSET_PLANS" on page 3-46.

USER_SQLSET_REFERENCES

USER_SQLSET_REFERENCES describes whether or not the SQL tuning sets owned by the current user are active. Its columns (except for SQLSET_OWNER) are the same as those in "ALL_SQLSET_REFERENCES" on page 3-49.

USER_SQLSET_STATEMENTS

USER_SQLSET_STATEMENTS displays information about the SQL statements, along with their statistics, that form the SQL tuning sets owned by the current user. Its columns (except for SQLSET_OWNER) are the same as those in "ALL_SQLSET_STATEMENTS" on page 3-49.

USER_SQLTUNE_BINDS

USER_SQLTUNE_BINDS displays the bind values associated with the tuned SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_BINDS.

See Also: "DBA_SQLTUNE_BINDS" on page 5-39

USER_SQLTUNE_PLANS

USER_SQLTUNE_PLANS displays information about the execution plans generated for the SQL statements owned by the current user during a SQL tuning session. Its columns are the same as those in DBA_SQLTUNE_PLANS.

See Also: "DBA_SQLTUNE_PLANS" on page 5-40

USER_SQLTUNE_RATIONALE_PLAN

USER_SQLTUNE_RATIONALE_PLAN displays the association between rationales and operations in the execution plan of the SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_RATIONALE_PLAN.

See Also: "DBA_SQLTUNE_RATIONALE_PLAN" on page 5-41

USER_SQLTUNE_STATISTICS

USER_SQLTUNE_STATISTICS displays statistics associated with the SQL statements owned by the current user. Its columns are the same as those in DBA_SQLTUNE_STATISTICS.

See Also: "DBA_SQLTUNE_STATISTICS" on page 5-42

USER_STAT_EXTENSIONS

USER_STAT_EXTENSIONS displays information about the optimizer statistics extensions owned by the current user. Its columns (except for OWNER) are the same as those in ALL_STAT_EXTENSIONS.

See Also: "ALL_STAT_EXTENSIONS" on page 3-51

USER_STORED_SETTINGS

USER_STORED_SETTINGS lists information about the persistent parameter settings for stored PL/SQL units, but only shows information about PL/SQL units owned by the current user. USER_STORED_SETTINGS does not display the OWNER column. The rest of its columns are the same as those in "ALL_STORED_SETTINGS" on page 3-51.

USER_SUBPART_COL_STATISTICS

USER_SUBPART_COL_STATISTICS provides column statistics and histogram information for subpartitions of subpartitioned objects owned by the current user. Its columns are the same as those in "ALL_STORED_SETTINGS" on page 3-51.

USER_SUBPART_HISTOGRAMS

USER_SUBPART_HISTOGRAMS lists actual histogram data (end-points per histogram) for histograms on table subpartitions owned by the current user. Its columns are the same as those in "ALL_SUBPART_HISTOGRAMS" on page 3-61.

USER_SUBPART_KEY_COLUMNS

USER_SUBPART_KEY_COLUMNS lists subpartitioning key columns for composite-partitioned tables (and local indexes on composite-partitioned tables) owned by the current user. Its columns are the same as those in ALL_SUBPART_KEY_COLUMNS.

See Also: "ALL_SUBPART_KEY_COLUMNS" on page 3-61

USER_SUBPARTITION_TEMPLATES

USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. Its columns (except for USER_NAME) are the same as those in ALL_SUBPARTITION_TEMPLATES.

See Also: "ALL_SUBPARTITION_TEMPLATES" on page 3-62

USER_SUBSCR_REGISTRATIONS

USER_SUBSCR_REGISTRATIONS displays information about the subscription registrations owned by the current user. Its columns are the same as those in DBA_SUBSCR_REGISTRATIONS.

See Also: "DBA_SUBSCR_REGISTRATIONS" on page 5-52

USER_SUBSCRIBED_COLUMNS

USER_SUBSCRIBED_COLUMNS describes the columns of source tables to which the current user has subscribed. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in DBA_SUBSCRIBED_COLUMNS.

See Also: "DBA_SUBSCRIBED_COLUMNS" on page 5-53

USER_SUBSCRIBED_TABLES

USER_SUBSCRIBED_TABLES describes the source tables to which the current user has subscribed. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in DBA_SUBSCRIBED_TABLES.

See Also: "DBA_SUBSCRIBED_TABLES" on page 5-54

USER_SUBSCRIPTIONS

USER_SUBSCRIPTIONS describes the subscriptions created by the current user. This view is intended for use by Change Data Capture subscribers. Its columns are the same as those in DBA_SUBSCRIPTIONS.

See Also: "DBA_SUBSCRIPTIONS" on page 5-54

USER_SYNONYMS

USER_SYNONYMS describes the private synonyms (synonyms owned by the current user). Its columns (except for OWNER) are the same as those in ALL_SYNONYMS.

See Also: "ALL_SYNONYMS" on page 3-64

USER_SYS_PRIVS

USER_SYS_PRIVS describes system privileges granted to the current user.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(30)		Name of the user, or PUBLIC
PRIVILEGE	VARCHAR2(40)	NOT NULL	System privilege
ADMIN_OPTION	VARCHAR2(3)		Indicates whether the grant was with the ADMIN option (YES) or not (NO)

USER_TAB_COL_STATISTICS

USER_TAB_COL_STATISTICS contains column statistics and histogram information extracted from "USER_TAB_COLUMNS" on page 5-131. Its columns are the same as those in "ALL_TAB_COL_STATISTICS" on page 3-65.

USER_TAB_COLS

USER_TAB_COLS describes the columns of the tables, views, and clusters owned by the current user. This view differs from "USER_TAB_COLUMNS" on page 5-131 in that hidden columns are not filtered out. Its columns (except for OWNER) are the same as those in "ALL_TAB_COLS" on page 3-66. To gather statistics for this view, use the ANALYZE SQL statement.

USER_TAB_COLUMNS

USER_TAB_COLUMNS describes the columns of the tables, views, and clusters owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_TAB_COLUMNS" on page 3-68. To gather statistics for this view, use the ANALYZE SQL statement.

USER_TAB_COMMENTS

USER_TAB_COMMENTS displays comments on the tables and views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TAB_COMMENTS.

See Also: "ALL_TAB_COMMENTS" on page 3-69

USER_TAB_HISTGRM_PENDING_STATS

USER_TAB_HISTGRM_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_TAB_HISTGRM_PENDING_STATS" on page 3-70.

USER_TAB_HISTOGRAMS

USER_TAB_HISTOGRAMS describes histograms on columns of tables owned by the current user. Its columns are the same as those in "ALL_TAB_HISTOGRAMS" on page 3-70.

USER_TAB_MODIFICATIONS

USER_TAB_MODIFICATIONS describes modifications to all tables owned by the current user that have been modified since the last time statistics were gathered on the tables. Its columns are the same as those in "ALL_TAB_MODIFICATIONS" on page 3-71.

Note: This view is populated only for tables with the MONITORING attribute. It is intended for statistics collection over a long period of time. For performance reasons, the Oracle Database does not populate this view immediately when the actual modifications occur. Run the FLUSH_DATABASE_MONITORING_INFO procedure in the DBMS_STATS PL/SQL package to populate this view with the latest information. The ANALYZE_ANY system privilege is required to run this procedure.

USER_TAB_PARTITIONS

USER_TAB_PARTITIONS describes partition-level partitioning information, partition storage parameters, and partition statistics determined by ANALYZE statements for all partitions owned by the current user. Its columns are the same as those in "ALL_TAB_PARTITIONS" on page 3-72.

USER_TAB_PENDING_STATS

USER_TAB_PENDING_STATS describes pending statistics for tables, partitions, and subpartitions owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_TAB_PENDING_STATS" on page 3-73.

USER_TAB_PRIVS

USER_TAB_PRIVS describes the object grants for which the current user is the object owner, grantor, or grantee. Its columns are the same as those in DBA_TAB_PRIVS.

See Also: "DBA_TAB_PRIVS" on page 5-57

USER_TAB_PRIVS_MADE

USER_TAB_PRIVS_MADE describes the object grants for which the current user is the object owner. Its columns (except for OWNER) are the same as those in ALL_TAB_PRIVS_MADE.

See Also: "ALL_TAB_PRIVS_MADE" on page 3-74

USER_TAB_PRIVS_RECD

USER_TAB_PRIVS_RECD describes the object grants for which the current user is the grantee. Its columns (except for GRANTEE) are the same as those in ALL_TAB_PRIVS_RECD.

See Also: "ALL_TAB_PRIVS_RECD" on page 3-74

USER_TAB_STAT_PREFS

USER_TAB_STAT_PREFS provides information about statistics preferences for tables owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_TAB_STAT_PREFS" on page 3-75.

USER_TAB_STATISTICS

USER_TAB_STATISTICS displays optimizer statistics for the tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TAB_STATISTICS.

See Also: "ALL_TAB_STATISTICS" on page 3-75

USER_TAB_STATS_HISTORY

USER_TAB_STATS_HISTORY provides a history of table statistics modifications for all tables owned by the current user. Its columns (except for OWNER) are the same as those in "ALL_TAB_STATS_HISTORY" on page 3-76.

USER_TAB_SUBPARTITIONS

USER_TAB_SUBPARTITIONS describes, for each table subpartition owned by the current user, the subpartition name, name of the table and partition to which it belongs, and its storage attributes. Its columns are the same as those in "ALL_TAB_SUBPARTITIONS" on page 3-77.

Note: Statistics are not collected on a per-subpartition basis.

USER_TABLES

USER_TABLES describes the relational tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TABLES. To gather statistics for this view, use the ANALYZE SQL statement.

See Also: "ALL_TABLES" on page 3-78

USER_TABLESPACES

USER_TABLESPACES describes the tablespaces accessible to the current user. Its columns (except for PLUGGED_IN) are the same as those in DBA_TABLESPACES.

See Also: "DBA_TABLESPACES" on page 5-58

USER_TRANSFORMATIONS

USER_TRANSFORMATIONS displays information about the transformations owned by the current user. Its columns (except for OWNER) are the same as those in DBA_TRANSFORMATIONS.

See Also: "DBA_TRANSFORMATIONS" on page 5-61

USER_TRIGGER_COLS

USER_TRIGGER_COLS describes the use of columns in the triggers owned by the current user and in triggers on tables owned by the current user. Its columns are the same as those in ALL_TRIGGER_COLS.

See Also: "ALL_TRIGGER_COLS" on page 3-80

USER_TRIGGER_ORDERING

USER_TRIGGER_ORDERING describes the triggers owned by the current user that have FOLLOWS or PRECEDES ordering. Its columns (except for TRIGGER_OWNER) are the same as those in ALL_TRIGGER_ORDERING.

See Also: "ALL_TRIGGER_ORDERING" on page 3-81

USER_TRIGGERS

USER_TRIGGERS describes the triggers owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TRIGGERS.

See Also: "ALL_TRIGGERS" on page 3-82

USER_TS_QUOTAS

USER_TS_QUOTAS contains information about tablespace quotas for the current user. Its columns are the same as those in "DBA_TS_QUOTAS" on page 5-62.

USER_TUNE_MVIEW

USER_TUNE_MVIEW displays the result of executing the DBMS_ADVISOR.TUNE_MVIEW procedure. Its columns (except for OWNER) are the same as those in DBA_TUNE_MVIEW.

See Also: "DBA_TUNE_MVIEW" on page 5-64

USER_TYPE_ATTRS

USER_TYPE_ATTRS describes the attributes of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_ATTRS.

See Also: "ALL_TYPE_ATTRS" on page 3-83

USER_TYPE_METHODS

USER_TYPE_METHODS describes the methods of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_METHODS.

See Also: "ALL_TYPE_METHODS" on page 3-84

USER_TYPE_VERSIONS

USER_TYPE_VERSIONS describes the versions of the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPE_VERSIONS.

See Also: "ALL_TYPE_VERSIONS" on page 3-84

USER_TYPES

USER_TYPES describes the object types owned by the current user. Its columns (except for OWNER) are the same as those in ALL_TYPES.

See Also: "ALL_TYPES" on page 3-85

USER_UNUSED_COL_TABS

USER_UNUSED_COL_TABS describes the tables owned by the current user that contain unused columns. Its columns (except for OWNER) are the same as those in ALL_UNUSED_COL_TABS.

See Also: "ALL_UNUSED_COL_TABS" on page 3-86

USER_UPDATABLE_COLUMNS

USER_UPDATABLE_COLUMNS describes columns in a join view that can be updated by the current user, subject to appropriate privileges. Its columns are the same as those in "ALL_UPDATABLE_COLUMNS" on page 3-86.

See Also: *Oracle Database Concepts* for information on updatable join views

USER_USERS

USER_USERS describes the current user. Its columns (except for PASSWORD, PROFILE, PASSWORD_VERSIONS, and EDITIONS_ENABLED) are the same as those in DBA_USERS.

See Also: "DBA_USERS" on page 5-65

USER_USTATS

USER_USTATS describes the user-defined statistics collected on the tables and indexes owned by the current user. Its columns are the same as those in ALL_USTATS.

See Also: "ALL_USTATS" on page 3-87

USER_VARRAYS

USER_VARRAYS describes the varrays owned by the current user. Its columns (except for OWNER) are the same as those in ALL_VARRAYS.

See Also: "ALL_VARRAYS" on page 3-88

USER_VIEWS

USER_VIEWS describes the views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_VIEWS.

See Also: "ALL_VIEWS" on page 3-88

USER_WARNING_SETTINGS

USER_WARNING_SETTINGS displays information about the warning parameter settings for the objects owned by the current user. Its columns (except for OWNER) are the same as those in ALL_WARNING_SETTINGS.

See Also: "ALL_WARNING_SETTINGS" on page 3-89

USER_XML_INDEXES

USER_XML_INDEXES describes the XML indexes owned by the current user. Its columns (except for INDEX_OWNER) are the same as those in ALL_XML_INDEXES.

See Also: "ALL_XML_INDEXES" on page 3-90

USER_XML_SCHEMAS

USER_XML_SCHEMAS describes the registered XML schemas owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_SCHEMAS.

See Also: "ALL_XML_SCHEMAS" on page 3-91

USER_XML_TAB_COLS

USER_XML_TAB_COLS describes the columns of the XML tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_TAB_COLS.

See Also: "ALL_XML_TAB_COLS" on page 3-91

USER_XML_TABLES

USER_XML_TABLES describes the XML tables owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_TABLES.

See Also: "ALL_XML_TABLES" on page 3-92

USER_XML_VIEW_COLS

USER_XML_VIEW_COLS describes the columns of the XML views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_VIEW_COLS.

See Also: "ALL_XML_VIEW_COLS" on page 3-93

USER_XML_VIEWS

USER_XML_VIEWS describes the XML views owned by the current user. Its columns (except for OWNER) are the same as those in ALL_XML_VIEWS.

See Also: "ALL_XML_VIEWS" on page 3-93

XS_SESSION_ROLES

XS_SESSION_ROLES describes the enabled roles in the current lightweight user session for the current request.

Column	Datatype	NULL	Description
ROLE	VARCHAR2(1025)		Name of the role
UUID	VARCHAR2(33)		UUID of the role
DBID	NUMBER		Database ID of the role
FLAGS	NUMBER		Status flags

Part III

Dynamic Performance Views

This part describes the dynamic performance views, which are often referred to as V$ views. This part contains the following chapters:

6

Dynamic Performance (V$) Views: V$ACCESS to V$HVMASTER_INFO

This chapter describes the first set (in alphabetical order) of dynamic performance views. The remaining dynamic performance views appear in alphabetical order in Chapter 7 through Chapter 8.

This chapter contains the following topics:

- About Dynamic Performance Views
- Dynamic Performance View Descriptions

About Dynamic Performance Views

Oracle contains a set of underlying views that are maintained by the database server and accessible to the database administrator user SYS. These views are called **dynamic performance views** because they are continuously updated while a database is open and in use, and their contents relate primarily to performance.

Although these views appear to be regular database tables, they are not. These views provide data on internal disk structures and memory structures. You can select from these views, but you can never update or alter them.

Note:

- You can query the dynamic performance views to extract information from them. However, only simple queries are supported. If sorts, joins, GROUP BY clauses and the like are needed, then you should copy the information from each V$ view into a table (for example, using a CREATE TABLE ... AS SELECT statement), and then query from those tables.
- Because the information in the V$ views is dynamic, read consistency is not guaranteed for SELECT operations on these views.

The catalog.sql script contains definitions of the views and public synonyms for the dynamic performance views. You must run catalog.sql to create these views and synonyms. After installation, only user SYS or anyone with SYSDBA role has access to the dynamic performance tables.

V$ Views

The actual dynamic performance views are identified by the prefix V_$. Public synonyms for these views have the prefix V$. Database administrators and other users should access only the V$ objects, not the V_$ objects.

The dynamic performance views are used by Oracle Enterprise Manager, which is the primary interface for accessing information about system performance. After an instance is started, the V$ views that read from memory are accessible. Views that read data from disk require that the database be mounted, and some require that the database be open.

GV$ Views

For almost every V$ view described in this chapter, Oracle has a corresponding GV$ (global V$) view. In Real Application Clusters, querying a GV$ view retrieves the V$ view information from all qualified instances. In addition to the V$ information, each GV$ view contains an extra column named INST_ID of datatype NUMBER. The INST_ID column displays the instance number from which the associated V$ view information was obtained. The INST_ID column can be used as a filter to retrieve V$ information from a subset of available instances. For example, the following query retrieves the information from the V$LOCK view on instances 2 and 5:

```
SQL> SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5;
```

See Also: *Oracle Real Application Clusters Installation and Configuration Guide*

Dynamic Performance View Descriptions

The remainder of this chapter describes the dynamic performance views in alphabetical order.

V$ACCESS

V$ACCESS displays information about locks that are currently imposed on library cache objects. The locks are imposed to ensure that they are not aged out of the library cache while they are required for SQL execution.

Column	Datatype	Description
SID	NUMBER	Session number that is accessing an object
OWNER	VARCHAR2(64)	Owner of the object
OBJECT	VARCHAR2(1000)	Name of the object
TYPE	VARCHAR2(24)	Type identifier for the object

V$ACTIVE_INSTANCES

V$ACTIVE_INSTANCES displays the mapping between instance names and instance numbers for all instances that have the database currently mounted.

Column	Datatype	Description
INST_NUMBER	NUMBER	Instance number
INST_NAME	VARCHAR2(60)	Instance name

V$ACTIVE_SERVICES

V$ACTIVE_SERVICES displays information about the active services in the database.

Column	Datatype	Description
SERVICE_ID	NUMBER	Service ID
NAME	VARCHAR2(64)	Name of the service
NAME_HASH	NUMBER	Service name hash
NETWORK_NAME	VARCHAR2(512)	Network name
CREATION_DATE	DATE	Creation date
CREATION_DATE_HASH	NUMBER	Creation date hash
GOAL	VARCHAR2(12)	Service workload management goal: ▪ NONE ▪ SERVICE_TIME ▪ THROUGHPUT
DTP	VARCHAR2(1)	Indicates whether the service is for DTP or distributed transactions including XA transactions (Y) or not (N)
BLOCKED	VARCHAR2(3)	Indicates whether a service on the specified instance is blocked from accepting new connections altogether (YES) or not (NO). If a service is blocked, then all connections will be directed to other instances (if any) that are hosting the desired service.
AQ_HA_NOTIFICATION	VARCHAR2(3)	Indicates whether AQ notifications are sent for HA events (YES) or not (NO)
CLB_GOAL	VARCHAR2(5)	Connection load balancing goal used with statistics that are sent to the listeners to determine how new connections are distributed: ▪ LONG ▪ SHORT

V$ACTIVE_SESS_POOL_MTH

V$ACTIVE_SESS_POOL_MTH displays available active session pool resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the active session pool resource allocation method

V$ACTIVE_SESSION_HISTORY

V$ACTIVE_SESSION_HISTORY displays sampled session activity in the database. It contains snapshots of active database sessions taken once a second. A database session is considered active if it was on the CPU or was waiting for an event that didn't belong to the Idle wait class. Refer to the V$EVENT_NAME view for more information on wait classes.

This view contains one row for each active session per sample and returns the latest session sample rows first. A majority of the columns describing the session in the active session history are present in the V$SESSION view.

Column	Datatype	Description
SAMPLE_ID	NUMBER	ID of the sample
SAMPLE_TIME	TIMESTAMP(3)	Time at which the sample was taken
SESSION_ID	NUMBER	Session identifier; maps to V$SESSION.SID

Column	Datatype	Description
SESSION_SERIAL#	NUMBER	Session serial number (used to uniquely identify a session's objects); maps to V$SESSION.SERIAL#
SESSION_TYPE	VARCHAR2(10)	Session type: - FOREGROUND - BACKGROUND
FLAGS	NUMBER	Reserved for future use
USER_ID	NUMBER	Oracle user identifier; maps to V$SESSION.USER#
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement that the session was executing at the time of sampling
SQL_CHILD_NUMBER	NUMBER	Child number of the SQL statement that the session was executing at the time of sampling
SQL_OPCODE	NUMBER	Indicates what phase of operation the SQL statement was in; maps to V$SESSION.COMMAND **See Also:** "V$SESSION" on page 8-8 for information on interpreting this column
FORCE_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to FORCE
TOP_LEVEL_SQL_ID	VARCHAR2(13)	SQL identifier of the top level SQL statement
TOP_LEVEL_SQL_OPCODE	NUMBER	Indicates what phase of operation the top level SQL statement was in
SQL_PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for the cursor. This information might not be available for all session samples. V$SESSION does not contain this information.
SQL_PLAN_LINE_ID	NUMBER	SQL plan line ID
SQL_PLAN_OPERATION	VARCHAR2(30)	Plan operation name
SQL_PLAN_OPTIONS	VARCHAR2(30)	Plan operation options
SQL_EXEC_ID	NUMBER	SQL execution identifier
SQL_EXEC_START	DATE	Time when the execution of the SQL started
PLSQL_ENTRY_OBJECT_ID	NUMBER	Object ID of the top-most PL/SQL subprogram on the stack; NULL if there is no PL/SQL subprogram on the stack
PLSQL_ENTRY_SUBPROGRAM_ID	NUMBER	Subprogram ID of the top-most PL/SQL subprogram on the stack; NULL if there is no PL/SQL subprogram on the stack
PLSQL_OBJECT_ID	NUMBER	Object ID of the currently executing PL/SQL subprogram; NULL if executing SQL
PLSQL_SUBPROGRAM_ID	NUMBER	Subprogram ID of the currently executing PL/SQL object; NULL if executing SQL
QC_INSTANCE_ID	NUMBER	Query coordinator instance ID. This information is only available if the sampled session is a parallel query slave. For all other sessions, the value is 0.
QC_SESSION_ID	NUMBER	Query coordinator session ID. This information is only available if the sampled session is a parallel query slave. For all other sessions, the value is 0.
QC_SESSION_SERIAL#	NUMBER	Query coordinator session serial number. This information is only available if the sampled session is a parallel query slave. For all other sessions, the value is 0.
EVENT	VARCHAR2(64)	If SESSION_STATE = WAITING, then the event for which the session was waiting for at the time of sampling. If SESSION_STATE = ON CPU, then this column is NULL. **See Also:** Appendix C, "Oracle Wait Events"
EVENT_ID	NUMBER	Identifier of the resource or event for which the session is waiting or for which the session last waited. Interpretation is similar to that of the EVENT column.

Column	Datatype	Description
EVENT#	NUMBER	Number of the resource or event for which the session is waiting or for which the session last waited. Interpretation is similar to that of the EVENT column.
SEQ#	NUMBER	Sequence number that uniquely identifies the wait (incremented for each wait)
P1TEXT	VARCHAR2(64)	Text of the first additional parameter
P1	NUMBER	First additional parameter
P2TEXT	VARCHAR2(64)	Text of the second additional parameter
P2	NUMBER	Second additional parameter
P3TEXT	VARCHAR2(64)	Text of the third additional parameter
P3	NUMBER	Third additional parameter
WAIT_CLASS	VARCHAR2(64)	Wait class name of the event for which the session was waiting at the time of sampling. Interpretation is similar to that of the EVENT column. Maps to V$SESSION.WAIT_CLASS.
WAIT_CLASS_ID	NUMBER	Wait class identifier of the event for which the session was waiting at the time of sampling. Interpretation is similar to that of the EVENT column. Maps to V$SESSION.WAIT_CLASS_ID.
WAIT_TIME	NUMBER	Total wait time for the event for which the session last waited if the session was on the CPU when sampled; 0 if the session was waiting at the time of sampling Note: Whether or not WAIT_TIME = 0 is what is useful to find the SESSION_STATE at the time of sampling, rather than the actual value of WAIT_TIME itself. Maps to V$SESSION.WAIT_TIME.
SESSION_STATE	VARCHAR2(7)	Session state: ■ WAITING ■ ON CPU
TIME_WAITED	NUMBER	If SESSION_STATE = WAITING, then the time that the session actually spent waiting for that EVENT. This column is set for waits that were in progress at the time the sample was taken. If a wait event lasted for more than a second and was caught waiting in more than one session sample row, then the actual time spent waiting for that wait event will be populated in the last of those session sample rows. At any given time, this information will not be available for the latest session sample.
BLOCKING_SESSION_STATUS	VARCHAR2(11)	Status of the blocking session: ■ VALID ■ NO HOLDER ■ GLOBAL ■ NOT IN WAIT ■ UNKNOWN
BLOCKING_SESSION	NUMBER	Session identifier of the blocking session. Populated only if the blocker is on the same instance and the session was waiting for enqueues or a "buffer busy" wait. Maps to V$SESSION.BLOCKING_SESSION.
BLOCKING_SESSION_SERIAL#	NUMBER	Serial number of the blocking session
CURRENT_OBJ#	NUMBER	Object ID of the object that the session is referencing. This information is only available if the session was waiting for application, cluster, concurrency, and user I/O wait events. Maps to V$SESSION.ROW_WAIT_OBJ#.
CURRENT_FILE#	NUMBER	File number of the file containing the block that the session is referencing. This information is only available if the session was waiting for cluster, concurrency, and user I/O wait events. Maps to V$SESSION.ROW_WAIT_FILE#.

Column	Datatype	Description
CURRENT_BLOCK#	NUMBER	ID of the block that the session is referencing. This information is only available if the session was waiting for cluster, concurrency, and user I/O wait events. Maps to V$SESSION.ROW_WAIT_BLOCK#.
CURRENT_ROW#	NUMBER	Row identifier that the session is referencing. This information is only available if the session was waiting for cluster, concurrency, and user I/O wait events. Maps to V$SESSION.ROW_WAIT_ROW#.
CONSUMER_GROUP_ID	NUMBER	Consumer group ID
XID	RAW(8)	Transaction ID that the session was working on at the time of sampling. V$SESSION does not contain this information.
REMOTE_INSTANCE#	NUMBER	Remote instance identifier that will serve the block that this session is waiting for. This information is only available if the session was waiting for cluster events.
IN_CONNECTION_MGMT	VARCHAR2(1)	Indicates whether the session was doing connection management at the time of sampling (Y) or not (N)
IN_PARSE	VARCHAR2(1)	Indicates whether the session was parsing at the time of sampling (Y) or not (N)
IN_HARD_PARSE	VARCHAR2(1)	Indicates whether the session was hard parsing at the time of sampling (Y) or not (N)
IN_SQL_EXECUTION	VARCHAR2(1)	Indicates whether the session was executing SQL statements at the time of sampling (Y) or not (N)
IN_PLSQL_EXECUTION	VARCHAR2(1)	Indicates whether the session was executing PL/SQL at the time of sampling (Y) or not (N)
IN_PLSQL_RPC	VARCHAR2(1)	Indicates whether the session was executing inbound PL/SQL RPC calls at the time of sampling (Y) or not (N)
IN_PLSQL_COMPILATION	VARCHAR2(1)	Indicates whether the session was compiling PL/SQL at the time of sampling (Y) or not (N)
IN_JAVA_EXECUTION	VARCHAR2(1)	Indicates whether the session was executing Java at the time of sampling (Y) or not (N)
IN_BIND	VARCHAR2(1)	Indicates whether the session was doing bind operations at the time of sampling (Y) or not (N)
IN_CURSOR_CLOSE	VARCHAR2(1)	Indicates whether the session was closing a cursor at the time of sampling (Y) or not (N)
SERVICE_HASH	NUMBER	Hash that identifies the Service; maps to V$ACTIVE_SERVICES.NAME_HASH
PROGRAM	VARCHAR2(64)	Name of the operating system program
MODULE	VARCHAR2(48)	Name of the executing module when sampled, as set by the DBMS_APPLICATION_INFO.SET_MODULE procedure
ACTION	VARCHAR2(32)	Name of the executing module when sampled, as set by the DBMS_APPLICATION_INFO.SET_ACTION procedure
CLIENT_ID	VARCHAR2(64)	Client identifier of the session; maps to V$SESSION.CLIENT_IDENTIFIER

V$ADVISOR_PROGRESS

V$ADVISOR_PROGRESS displays information about the progress of advisor execution.

Column	Datatype	Description
SID	NUMBER	Session ID
SERIAL#	NUMBER	Session serial number
USERNAME	VARCHAR2(30)	Oracle user name
OPNAME	VARCHAR2(64)	Operation name
ADVISOR_NAME	VARCHAR2(64)	Advisor name

Column	Datatype	Description
TASK_ID	NUMBER	Task ID
TARGET_DESC	VARCHAR2(32)	Description of the target of the advisor
SOFAR	NUMBER	Amount of work done so far
TOTALWORK	NUMBER	Total work to be done
UNITS	VARCHAR2(32)	Units that the work is measured in
BENEFIT_SOFAR	NUMBER	Benefit obtained so far
BENEFIT_MAX	NUMBER	Estimate of maximum benefit that could be obtained
FINDINGS	NUMBER	Number of findings so far
RECOMMENDATIONS	NUMBER	Number of recommendations so far
TIME_REMAINING	NUMBER	Estimate of time remaining for the completion of the task (in seconds)
START_TIME	DATE	Start time of the task
LAST_UPDATE_TIME	DATE	Last time progress was posted
ELAPSED_SECONDS	NUMBER	Elapsed time so far
ADVISOR_METRIC1	NUMBER	Value of the advisor-specific metric
METRIC1_DESC	VARCHAR2(64)	Description of the advisor-specific metric
EXECUTION_TYPE	VARCHAR2(64)	???

V$ALERT_TYPES

V$ALERT_TYPES displays information about server alert types.

Column	Datatype	Description
REASON_ID	NUMBER	ID of the alert reason
OBJECT_TYPE	VARCHAR2(64)	Object type
TYPE	VARCHAR2(9)	Alert type: ■ Stateful ■ Stateless
GROUP_NAME	VARCHAR2(64)	Group name
SCOPE	VARCHAR2(8)	Scope: ■ Database ■ Instance
INTERNAL_METRIC_CATEGORY	VARCHAR2(64)	Internal metric category
INTERNAL_METRIC_NAME	VARCHAR2(64)	Internal metric name

V$AQ

V$AQ displays statistics for the queues in the database.

Column	Datatype	Description
QID	NUMBER	Unique queue identifier
WAITING	NUMBER	Number of messages in the queue in the state 'WAITING'
READY	NUMBER	Number of messages in the queue in the state 'READY'
EXPIRED	NUMBER	Number of messages in the queue the state 'EXPIRED'
TOTAL_WAIT	NUMBER	Total wait time of all 'READY' messages in the queue

Column	Datatype	Description
AVERAGE_WAIT	NUMBER	Average wait time of 'READY' messages in the queue

V$ARCHIVE

V$ARCHIVE displays information about redo log files in need of archiving. Each row displays information for one thread. This information is also available in V$LOG. Oracle recommends that you use V$LOG.

See Also: "V$LOG" on page 7-15

Column	Datatype	Description
GROUP#	NUMBER	Log file group number
THREAD#	NUMBER	Log file thread number
SEQUENCE#	NUMBER	Log file sequence number
ISCURRENT	VARCHAR2(3)	This is the current online redo log
CURRENT	VARCHAR2(3)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in ISCURRENT.
FIRST_CHANGE#	NUMBER	First SCN stored in the current log

V$ARCHIVE_DEST

V$ARCHIVE_DEST displays, for the current instance, all of the destinations in the Data Guard configuration, including each destination's current value, mode, and status.

Column	Datatype	Description
DEST_ID	NUMBER	Log archive destination parameter identifier (1 to 10)
DEST_NAME	VARCHAR2(256)	Log archive destination parameter name
STATUS	VARCHAR2(9)	Identifies the current status of the destination: ■ VALID - Initialized and available ■ INACTIVE - No destination information ■ DEFERRED - Manually disabled by the user ■ ERROR - Error during open or copy ■ DISABLED - Disabled after error ■ BAD PARAM - Parameter has errors ■ ALTERNATE - Destination is in an alternate state ■ FULL - Exceeded quota size for the destination
BINDING	VARCHAR2(9)	Specifies how failure will affect the archival operation: ■ MANDATORY - Successful archival is required ■ OPTIONAL - Successful archival is not required (depends on LOG_ARCHIVE_MIN_SUCCEED_DEST)
NAME_SPACE	VARCHAR2(7)	Identifies the scope of parameter setting: ■ SYSTEM - System definition ■ SESSION - Session definition
TARGET	VARCHAR2(7)	Specifies whether the archive destination is local or remote to the primary database: ■ PRIMARY - local ■ STANDBY - remote

Column	Datatype	Description
ARCHIVER	VARCHAR2(10)	Identifies the archiver process relative to the database where the query is issued: ■ ARC*n* ■ FOREGROUND ■ LGWR ■ RFS
SCHEDULE	VARCHAR2(8)	Indicates whether the archival of this destination is INACTIVE, PENDING, ACTIVE, or LATENT
DESTINATION	VARCHAR2(256)	Specifies the location where the archived redo logs are to be archived
LOG_SEQUENCE	NUMBER	Identifies the sequence number of the last archived redo log to be archived
REOPEN_SECS	NUMBER	Identifies the retry time (in seconds) after error
DELAY_MINS	NUMBER	Identifies the delay interval (in minutes) before the archived redo log is automatically applied to a standby database
MAX_CONNECTIONS	NUMBER	Maximum number of connections
NET_TIMEOUT	NUMBER	Number of seconds the log writer process will wait for status from the network server of a network operation issued by the log writer process
PROCESS	VARCHAR2(10)	Identifies the archiver process relative to the primary database, even if the query is issued on the standby database: ■ ARC*n* ■ FOREGROUND ■ LGWR
REGISTER	VARCHAR2(3)	Indicates whether the archived redo log is registered in the remote destination control file (YES) or not (NO). If the archived redo log is registered, it is available to log apply services.
FAIL_DATE	DATE	Date and time of last error
FAIL_SEQUENCE	NUMBER	Sequence number of the archived redo log being archived when the last error occurred
FAIL_BLOCK	NUMBER	Block number of the archived redo log being archived when the last error occurred
FAILURE_COUNT	NUMBER	Current number of contiguous archival operation failures that have occurred for the destination
MAX_FAILURE	NUMBER	Allows you to control the number of times log transport services will attempt to reestablish communication and resume archival operations with a failed destination
ERROR	VARCHAR2(256)	Displays the error text
ALTERNATE	VARCHAR2(256)	Alternate destination, if any
DEPENDENCY	VARCHAR2(256)	Reserved for future use
REMOTE_TEMPLATE	VARCHAR2(256)	Specifies the template to be used to derive the location to be recorded
QUOTA_SIZE	NUMBER	Destination quotas, expressed in bytes
QUOTA_USED	NUMBER	Size of all the archived redo logs currently residing on the specified destination
MOUNTID	NUMBER	Instance mount identifier
TRANSMIT_MODE	VARCHAR2(12)	Specifies network transmission mode: ■ SYNCHRONOUS ■ PARALLELSYNC ■ ASYNCHRONOUS
ASYNC_BLOCKS	NUMBER	Number of blocks specified for the ASYNC attribute
AFFIRM	VARCHAR2(3)	Specifies disk I/O mode

Column	Datatype	Description
TYPE	VARCHAR2(7)	Indicates whether the archived log destination definition is PUBLIC or PRIVATE. Only PUBLIC destinations can be modified at runtime using the ALTER SYSTEM SET or ALTER SESSION SET statements. By default, all archived log destinations are PUBLIC.
VALID_NOW	VARCHAR2(16)	Indicates whether the destination is valid right now for archival operations: ■ YES - Redo log type and database role for this destination are valid for the current database ■ WRONG VALID_TYPE - Redo log type specified for this destination is not valid for the current database role. For example, WRONG VALID_TYPE would be returned if a destination specified with the VALID_FOR=(STANDBY_LOGFILE,STANDBY_ROLE) attribute is running in the standby database role but does not have standby redo logs implemented. ■ WRONG VALID_ROLE - Database role specified for this destination is not the role in which the database is currently running. For example, the WRONG VALID_ROLE would be returned when a destination defined with the VALID_FOR=(ONLINE_LOGFILE,STANDBY_ROLE) attribute is running in the primary database role. ■ INACTIVE - Destination is inactive, probably due to an error
VALID_TYPE	VARCHAR2(15)	Redo log type or types that are valid for the destination: ■ ONLINE_LOGFILE ■ STANDBY_LOGFILE ■ ALL_LOGFILES
VALID_ROLE	VARCHAR2(12)	Database role or roles that are valid for the destination: ■ PRIMARY_ROLE ■ STANDBY_ROLE ■ ALL_ROLES
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name
VERIFY	VARCHAR2(3)	Indicates whether the value of the VERIFY attribute on the LOG_ARCHIVE_DEST_*n* parameter is verified (YES) or not verified (NO)
COMPRESSION	VARCHAR2(7)	Indicates whether network compression is ENABLED or DISABLED.

See Also:

- "LOG_ARCHIVE_DEST" on page 1-80 and "LOG_ARCHIVE_DEST_n" on page 1-81
- "LOG_ARCHIVE_DUPLEX_DEST" on page 1-86 and "LOG_ARCHIVE_DEST_STATE_n" on page 1-85
- "STANDBY_ARCHIVE_DEST" on page 1-155
- "LOG_ARCHIVE_MIN_SUCCEED_DEST" on page 1-89

V$ARCHIVE_DEST_STATUS

V$ARCHIVE_DEST_STATUS displays runtime and configuration information for the archived redo log destinations. The information in this view does not persist across an instance shutdown.

Column	Datatype	Description
DEST_ID	NUMBER	Identifies the log archive destination parameter (1 to 10)
DEST_NAME	VARCHAR2(256)	Log archive destination parameter name

Column	Datatype	Description
STATUS	VARCHAR2(9)	Current status of the destination: ■ VALID - Initialized and available ■ INACTIVE - No destination information ■ DEFERRED - Manually disabled by the user ■ ERROR - Error during open or copy ■ DISABLED - Disabled after error ■ BAD PARAM - Parameter has errors ■ ALTERNATE - Destination is in an alternate state ■ FULL - Exceeded quota size for the destination
TYPE	VARCHAR2(14)	Type of archival destination database: ■ LOCAL - Local to primary database ■ PHYSICAL - Physical standby ■ CROSS-INSTANCE - An instance of the primary ■ LOGICAL - Logical standby
DATABASE_MODE	VARCHAR2(15)	Current mode of the archival destination database: ■ STARTED - Instance started, not mounted ■ MOUNTED - Mounted ■ MOUNTED-STANDBY - Mounted standby ■ OPEN - Open read/write ■ OPEN_READ-ONLY - Open read-only
RECOVERY_MODE	VARCHAR2(23)	Current mode of media recovery at the archival destination database: ■ IDLE - Managed recovery is not active ■ MANUAL - Manual media recovery active ■ MANAGED - Managed recovery is active ■ MANAGED REAL TIME APPLY - Log apply services recover redo data from standby redo logs at the same time the logs are being written to, as opposed to recovering redo from archived redo logs when a log switch occurs
PROTECTION_MODE	VARCHAR2(20)	Indicates whether the database is protected: ■ MAXIMUM PROTECTION ■ MAXIMUM AVAILABILITY ■ RESYNCHRONIZATION ■ MAXIMUM PERFORMANCE ■ UNPROTECTED
DESTINATION	VARCHAR2(256)	Specifies the location where the redo data is to be archived
STANDBY_LOGFILE_COUNT	NUMBER	Indicates the total number of standby redo logs created on the standby database
STANDBY_LOGFILE_ACTIVE	NUMBER	Indicates the total number of standby redo logs on the standby database that are active and contain primary database online redo log information
ARCHIVED_THREAD#	NUMBER	Identifies the thread number of the most recent archived redo log received at the destination
ARCHIVED_SEQ#	NUMBER	Identifies the log sequence number of the most recent archived redo log received at the destination
APPLIED_THREAD#	NUMBER	Identifies the thread number of the most recent applied redo log received at the destination
APPLIED_SEQ#	NUMBER	Identifies the log sequence number of the most recent applied redo log received at the destination
ERROR	VARCHAR2(256)	Displays the error text
SRL	VARCHAR2(3)	Indicates whether standby redo logfiles are used on the standby database (YES) or not (NO)

Column	Datatype	Description
DB_UNIQUE_NAME	VARCHAR2(30)	Specifies the unique database name of the current instance that was defined with the DB_UNIQUE_NAME attribute on the LOG_ARCHIVE_DEST_*n* parameter
SYNCHRONIZATION_STATUS	VARCHAR2(22)	Possible values for this column are as follows: ■ CHECK CONFIGURATION - This database or destination does not support synchronization. ■ CHECK STANDBY REDO LOG - The standby redo log at this destination is configured improperly. ■ CHECK NETWORK - One or more instances of this database cannot send redo data to this destination. ■ DESTINATION HAS A GAP - This destination is missing redo data needed for synchronization with this database. ■ OK - This destination is synchronized with this database. ■ NOT AVAILABLE - Synchronization status is not available.
SYNCHRONIZED	VARCHAR2(3)	Possible values are: ■ YES - This destination is synchronized with the primary database. ■ NO - The destination is not synchronized with the primary database. ■ UNKNOWN - The synchronization status of this destination cannot be determined.

V$ARCHIVE_GAP

V$ARCHIVE_GAP displays information about archive gaps on a standby database. This view can be used to find out the current archive gap that is blocking recovery.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the missing archived redo log files. The number is 1 for a single instance. For Real Application Clusters, this column will contain different numbers.
LOW_SEQUENCE#	NUMBER	Lowest sequence number of the log files received on the standby system
HIGH_SEQUENCE#	NUMBER	Highest sequence number of the log files received on the standby system

V$ARCHIVE_PROCESSES

V$ARCHIVE_PROCESSES displays the state of the various ARCH processes for the instance.

Column	Datatype	Description
PROCESS	NUMBER	Identifier for the ARCH process for the instance, numbered from 0-9
STATUS	VARCHAR2(10)	Status of the ARCH process, displayed as a keyword. Possible values are: STOPPED, SCHEDULED, STARTING, ACTIVE, STOPPING, and TERMINATED.
LOG_SEQUENCE	NUMBER	This is the online redo log sequence number currently being archived, if STATE="BUSY"
STATE	VARCHAR2(4)	This is the current state of the ARCH process, displayed as a keyword. Possible keywords are IDLE or BUSY.

V$ARCHIVED_LOG

V$ARCHIVED_LOG displays archived log information from the control file, including archive log names. An archive log record is inserted after the online redo log is successfully archived or cleared (name column is NULL if the log was cleared). If the log is archived twice, there will be two archived log records with the same THREAD#, SEQUENCE#, and FIRST_CHANGE#, but with a different name. An archive log record is also inserted when an archive log is restored from a backup set or a copy and whenever a copy of a log is made with the RMAN COPY command.

Column	Datatype	Description
RECID	NUMBER	Archived log record ID
STAMP	NUMBER	Archived log record stamp
NAME	VARCHAR2(513)	Archived log file name. If set to NULL, either the log file was cleared before it was archived or an RMAN backup command with the "delete input" option was executed to back up archivelog all (RMAN> backup archivelog all delete input;).
DEST_ID	NUMBER	Original destination from which the archive log was generated. The value is 0 if the destination identifier is not available.
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when the log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when the log was written
RESETLOGS_ID	NUMBER	Resetlogs identifier associated with the archived redo log
FIRST_CHANGE#	NUMBER	First change number in the archived log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_CHANGE#	NUMBER	First change in the next log
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log (in blocks)
BLOCK_SIZE	NUMBER	Redo log block size. This is the logical block size of the archived log, which is the same as the logical block size of the online log from which the archived log was copied. The online log logical block size is a platform-specific value that is not adjustable by the user.
CREATOR	VARCHAR2(7)	Creator of the archivelog: ▪ ARCH - Archiver process ▪ FGRD - Foreground process ▪ RMAN - Recovery Manager ▪ SRMN - RMAN at standby ▪ LGWR - Logwriter process
REGISTRAR	VARCHAR2(7)	Registrar of the entry: ▪ RFS - Remote File Server process ▪ ARCH - Archiver process ▪ FGRD - Foreground process ▪ RMAN - Recovery manager ▪ SRMN - RMAN at standby ▪ LGWR - Logwriter process
STANDBY_DEST	VARCHAR2(3)	Indicates whether the entry is an archivelog destination (YES) or not (NO)
ARCHIVED	VARCHAR2(3)	Indicates whether the online redo log was archived (YES) or whether RMAN only inspected the log and created a record for future application of redo logs during recovery (NO). **See Also:** *Oracle Database Backup and Recovery Basics*.

Column	Datatype	Description
APPLIED	VARCHAR2(3)	Indicates whether the archivelog has been applied to its corresponding standby database (YES) or not (NO). The value is always NO for local destinations. This column is meaningful at the physical standby site for the ARCHIVED_LOG entries with REGISTRAR='RFS' (which means this log is shipped from the primary to the standby database). If REGISTRAR='RFS' and APPLIED is NO, then the log has arrived at the standby but has not yet been applied. If REGISTRAR='RFS' and APPLIED is YES, the log has arrived and been applied at the standby database. You can use this field to identify archivelogs that can be backed up and removed from disk.
DELETED	VARCHAR2(3)	Indicates whether an RMAN DELETE command has physically deleted the archived log file from disk, as well as logically removing it from the control file of the target database and from the recovery catalog (YES) or not (NO)
STATUS	VARCHAR2(1)	Status of the archived log: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the archiving completed
DICTIONARY_BEGIN	VARCHAR2(3)	Indicates whether the log contains the start of a LogMiner dictionary (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)	Indicates whether the log contains the end of a LogMiner dictionary (YES) or not (NO)
END_OF_REDO	VARCHAR2(3)	Indicates whether the archived redo log contains the end of all redo information from the primary database (YES) or not (NO)
BACKUP_COUNT	NUMBER	Indicates the number of times this file has been backed up. Values range from 0-15. If the file has been backed up more than 15 times, the value remains 15.
ARCHIVAL_THREAD#	NUMBER	Redo thread number of the instance that performed the archival operation. This column differs from the THREAD# column only when a closed thread is archived by another instance.
ACTIVATION#	NUMBER	Number assigned to the database instantiation
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
COMPRESSED	VARCHAR2(3)	This column is reserved for internal use only
FAL	VARCHAR2(3)	Indicates whether the archive log was generated as the result of a FAL request (YES) or not (NO)
END_OF_REDO_TYPE	VARCHAR2(10)	Possible values are as follows: ■ SWITCHOVER - Shows archived redo log files that are produced at the end of a switchover ■ TERMINAL - Shows archived redo log files produced after a failover ■ RESETLOGS - Shows online redo log files archived on the primary database after an ALTER DATABASE OPEN RESETLOGS statement is issued ■ ACTIVATION - Shows any log files archived on a physical standby database after an ALTER DATABASE ACTIVATE STANDBY DATABASE statement is issued ■ "empty string" - Any empty string implies that that the log is just a normal archival and was not archived due to any of the other events
BACKED_BY_VSS	VARCHAR2(3)	Whether or not the file has been backed up by Volume Shadow Copy Service (VSS). This column is for internal use only.

V$ASM_ALIAS

In an Automatic Storage Management instance, V$ASM_ALIAS displays one row for every alias present in every disk group mounted by the Automatic Storage Management instance. In a database instance, V$ASM_ALIAS displays no rows.

Column	Datatype	Description
NAME	VARCHAR2(70)	Automatic Storage Management alias or alias directory name
GROUP_NUMBER	NUMBER	Owning disk group number of the alias (foreign key to the V$ASM_DISKGROUP view)
FILE_NUMBER	NUMBER	Automatic Storage Management file number of the alias (foreign key to the V$ASM_FILE view)
FILE_INCARNATION	NUMBER	Automatic Storage Management file incarnation number for the alias
ALIAS_INDEX	NUMBER	Alias entry number for the alias
ALIAS_INCARNATION	NUMBER	Incarnation number for the parent of the alias
PARENT_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and an alias entry number in the low-order 24 bits (number of the directory containing the alias)
REFERENCE_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and an alias entry number in the low-order 24 bits (number of the directory describing the current entry)
ALIAS_DIRECTORY	VARCHAR2(1)	Indicates whether the alias is to a directory (Y) or to an Automatic Storage Management file (N)
SYSTEM_CREATED	VARCHAR2(1)	Indicates whether the alias is system created (Y) or user created (N)

V$ASM_ATTRIBUTE

V$ASM_ATTRIBUTE displays one row for each attribute defined. In addition to attributes specified by CREATE DISKGROUP and ALTER DISKGROUP statements, the view may show other attributes that are created automatically.

Column	Datatype	Description
NAME	VARCHAR2(256)	Full name of the attribute
VALUE	VARCHAR2(256)	Value of the attribute
GROUP_NUMBER	NUMBER	Number of the disk group in which this attribute exists (composite primary key)
ATTRIBUTE_INDEX	NUMBER	Number of this attribute in the disk group (composite primary key)
ATTRIBUTE_INCARNATION	NUMBER	Incarnation number for this attribute (composite primary key)
READ_ONLY	VARCHAR2(7)	Indicates whether the attribute is read-only (Y) or not (N)
SYSTEM_CREATED	VARCHAR2(7)	Indicates whether this is a system-created attribute (Y) or not (N)

V$ASM_CLIENT

V$ASM_CLIENT displays one row for each open ASM disk group.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group being used by the client database instance (foreign key to the V$ASM_DISKGROUP view)
INSTANCE_NAME	VARCHAR2(64)	Identifier for the database instance client
DB_NAME	VARCHAR2(8)	Unique database name of the database client instance

Column	Datatype	Description
STATUS	VARCHAR2(12)	Status of the client connection: ■ CONNECTED - Database instance client has an active connection to the Automatic Storage Management instance ■ DISCONNECTED - Database instance client normally ended its connection to the Automatic Storage Management instance ■ BROKEN - Connection with the database instance client terminated abnormally
SOFTWARE_VERSION	VARCHAR2(60)	Software version number of the database or Automatic Storage Management instance for the selected disk group connection
COMPATIBLE_VERSION	VARCHAR2(60)	Compatibility setting of the database or Automatic Storage Management instance for the selected disk group connection

V$ASM_DISK

In an Automatic Storage Management instance, V$ASM_DISK displays one row for every disk discovered by the Automatic Storage Management instance, including disks which are not part of any disk group. In a database instance, V$ASM_DISK only displays rows for disks in disk groups in use by the database instance.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group containing the disk (foreign key to the V$ASM_DISKGROUP view)
DISK_NUMBER	NUMBER	Number assigned to the disk within its disk group
COMPOUND_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and a disk number in the low-order 24 bits (for efficient access to the view)
INCARNATION	NUMBER	Incarnation number for the disk
MOUNT_STATUS	VARCHAR2(7)	Per-instance status of the disk relative to group mounts: ■ MISSING - Automatic Storage Management metadata indicates that the disk is known to be part of the Automatic Storage Management disk group but no disk in the storage system was found with the indicated name ■ CLOSED - Disk is present in the storage system but is not being accessed by Automatic Storage Management ■ OPENED - Disk is present in the storage system and is being accessed by Automatic Storage Management. This is the normal state for disks in a database instance which are part of a Disk Group being actively used by the instance. ■ CACHED - Disk is present in the storage system and is part of a disk group being accessed by the Automatic Storage Management instance. This is the normal state for disks in an Automatic Storage Management instance which are part of a mounted disk group. ■ IGNORED - Disk is present in the system but is ignored by ASM because of one of the following: – The disk is detected by the system library but is ignored because an ASM library discovered the same disk – ASM has determined that the membership claimed by the disk header is no longer valid ■ CLOSING - ASM is in the process of closing this disk

Column	Datatype	Description
HEADER_STATUS	VARCHAR2(12)	Per-instance status of the disk as seen by discovery: ■ UNKNOWN - Automatic Storage Management disk header has not been read ■ CANDIDATE - Disk is not part of a disk group and may be added to a disk group with the ALTER DISKGROUP statement ■ INCOMPATIBLE - Version number in the disk header is not compatible with the Automatic Storage Management software version ■ PROVISIONED - Disk is not part of a disk group and may be added to a disk group with the ALTER DISKGROUP statement. The PROVISIONED header status is different from the CANDIDATE header status in that PROVISIONED implies that an additional platform-specific action has been taken by an administrator to make the disk available for Automatic Storage Management. ■ MEMBER - Disk is a member of an existing disk group. No attempt should be made to add the disk to a different disk group. The ALTER DISKGROUP statement will reject such an addition unless overridden with the FORCE option. ■ FORMER - Disk was once part of a disk group but has been dropped cleanly from the group. It may be added to a new disk group with the ALTER DISKGROUP statement. ■ CONFLICT - Automatic Storage Management disk was not mounted due to a conflict ■ FOREIGN - Disk contains data created by an Oracle product other than ASM. This includes datafiles, logfiles, and OCR disks.
MODE_STATUS	VARCHAR2(7)	Global status about which kinds of I/O requests are allowed to the disk: ■ UNKNOWN - Automatic Storage Management disk mode is not known (typically the disk is not mounted) ■ ONLINE - Disk is online and operating normally. Reads and writes are attempted. ■ OFFLINE - Disk is offline and access to data is not permitted. Reads and writes are not attempted. An offline disk remains logically part of its disk group.
STATE	VARCHAR2(8)	Global state of the disk with respect to the disk group: ■ UNKNOWN - Automatic Storage Management disk state is not known (typically the disk is not mounted) ■ NORMAL - Disk is online and operating normally ■ ADDING - Disk is being added to a disk group, and is pending validation by all instances that have the disk group mounted ■ DROPPING - Disk has been manually taken offline and space allocation or data access for the disk halts. Rebalancing will commence to relocate data off the disks to other disks in the disk group. Upon completion of the rebalance, the disk is expelled from the group. ■ HUNG - Disk drop operation cannot continue because there is insufficient space to relocate the data from the disk being dropped ■ FORCING - Disk is being removed from the disk group without attempting to offload its data. The data will be recovered from redundant copies, where possible. ■ DROPPED - Disk has been fully expelled from the disk group
REDUNDANCY	VARCHAR2(7)	Hardware redundancy of the disk: ■ UNKNOWN ■ UNPROT ■ MIRROR ■ PARITY **Note:** This column is valid only if an ASMLIB is present that supports returning hardware redundancy information. This column is not related to the redundancy of the disk group of which the disk is a member.

Column	Datatype	Description
LIBRARY	VARCHAR2(64)	Name of the library that discovered the disk
OS_MB	NUMBER	Size of the disk (in megabytes) as reported by the host operating system
TOTAL_MB	NUMBER	Total capacity of the disk (in megabytes)
FREE_MB	NUMBER	Unused capacity of the disk (in megabytes)
NAME	VARCHAR2(30)	Name of the disk
FAILGROUP	VARCHAR2(30)	Name of the failure group containing the disk
LABEL	VARCHAR2(31)	Disk label portion of the name returned by discovery
PATH	VARCHAR2(256)	Operating system pathname portion of the name returned by discovery
UDID	VARCHAR2(64)	Unique Device ID portion of the name returned by discovery
PRODUCT	VARCHAR2(32)	Name of the manufacturer and the name of the product. All disks with the same product id will have the same performance and reliability characteristics.
CREATE_DATE	DATE	Date and time when the disk was added to the disk group
MOUNT_DATE	DATE	Date and time when the disk was mounted by the first instance
REPAIR_TIMER	NUMBER	Seconds remaining until the disk is automatically dropped (0 if not failed)
READS	NUMBER	Total number of I/O read requests for the disk
WRITES	NUMBER	Total number of I/O write requests for the disk
READ_ERRS	NUMBER	Total number of failed I/O read requests for the disk
WRITE_ERRS	NUMBER	Total number of failed I/O write requests for the disk
READ_TIME	NUMBER	Total I/O time (in hundredths of a second) for read requests for the disk if the TIMED_STATISTICS initialization parameter is set to true (0 if set to false)
WRITE_TIME	NUMBER	Total I/O time (in hundredths of a second) for write requests for the disk if the TIMED_STATISTICS initialization parameter is set to true (0 if set to false)
BYTES_READ	NUMBER	Total number of bytes read from the disk
BYTES_WRITTEN	NUMBER	Total number of bytes written to the disk
PREFERRED_READ	VARCHAR2(1)	Status of the preferred read failure group: ■ U - Disk group has no preferred read failure group ■ Y - Disk is a preferred read disk ■ N - Disk is not a preferred read disk For a disk group with one or more preferred read failure groups, if the disk is in one of the preferred read failure groups, the value of this column is Y; otherwise it is N.

Note: The GROUP_NUMBER and DISK_NUMBER columns will only be valid if the disk is part of a disk group which is currently mounted by the instance. Otherwise, GROUP_NUMBER will be 0, and DISK_NUMBER will be a unique value with respect to the other disks that also have a group number of 0.

V$ASM_DISK_IOSTAT

V$ASM_DISK_IOSTAT displays information about disk I/O statistics for each ASM client. If this view is queried from the database instance, only the rows for that instance are shown.

Column	Datatype	Description
INSTNAME	VARCHAR2(64)	Identifier for the DB/AVD instance client
DBNAME	VARCHAR2(8)	Unique database name (DB_UNIQUE_NAME)
GROUP_NUMBER	NUMBER	Number of the disk group containing the disk
DISK_NUMBER	NUMBER	Number assigned to the disk within its disk group
FAILGROUP	VARCHAR2(30)	Name of the failure group to which the disk belongs
READS	NUMBER	Total number of I/O read requests for the disk
WRITES	NUMBER	Total number of I/O write requests for the disk
READ_ERRS	NUMBER	Total number of failed I/O read requests for the disk
WRITE_ERRS	NUMBER	Total number of failed I/O write requests for the disk
READ_TIME	NUMBER	Total I/O time (in hundreths of a second) for read requests for the disk if the TIMED_STATISTICS initialization parameter is set to TRUE (0 if set to FALSE).
WRITE_TIME	NUMBER	Total I/O time (in hundreths of a second) for write requests for the disk if the TIMED_STATISTICS initialization parameter is set to TRUE (0 if set to FALSE)
BYTES_READ	NUMBER	Total number of bytes read from the disk
BYTES_WRITTEN	NUMBER	Total number of bytes written from the disk

V$ASM_DISK_STAT

V$ASM_DISK_STAT displays performance statistics in the same way that V$ASM_DISK does, but without performing discovery of new disks. This results in a less expensive operation. However, since discovery is not performed, the output of this view does not include any data about disks that are new to the system.

The columns for V$ASM_DISK_STAT are the same as those for V$ASM_DISK.

See Also: "V$ASM_DISK" on page 6-16

V$ASM_DISKGROUP

V$ASM_DISKGROUP displays one row for every ASM disk group discovered by the ASM instance on the node.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Cluster-wide number assigned to the disk group (primary key)
NAME	VARCHAR2(30)	Name of the disk group
SECTOR_SIZE	NUMBER	Physical block size (in bytes)
BLOCK_SIZE	NUMBER	Automatic Storage Management metadata block size (in bytes)
ALLOCATION_UNIT_SIZE	NUMBER	Size of the allocation unit (in bytes)
STATE	VARCHAR2(11)	State of the disk group relative to the instance: ■ CONNECTED - Disk group is in use by the database instance ■ BROKEN - Database instance lost connectivity to the Automatic Storage Management instance that mounted the disk group ■ UNKNOWN - Automatic Storage Management instance has never attempted to mount the disk group ■ DISMOUNTED - Disk group was cleanly dismounted by the Automatic Storage Management instance following a successful mount ■ MOUNTED - Instance is successfully serving the disk group to its database clients

Column	Datatype	Description
TYPE	VARCHAR2(6)	Redundancy type for the disk group: ■ EXTERN ■ NORMAL ■ HIGH
TOTAL_MB	NUMBER	Total capacity of the disk group (in megabytes)
FREE_MB	NUMBER	Unused capacity of the disk group (in megabytes)
REQUIRED_MIRROR_FREE_MB	NUMBER	Amount of space that is required to be available in a given disk group in order to restore redundancy after one or more disk failures. The amount of space displayed in this column takes mirroring effects into account.
USABLE_FILE_MB	NUMBER	Amount of free space that can be safely utilized taking mirroring into account and yet be able to restore redundancy after a disk failure
OFFLINE_DISKS	NUMBER	Number of disks in the disk group that are currently offline
COMPATIBILITY	VARCHAR2(60)	Minimum software version required for an ASM instance to mount this disk group
DATABASE_COMPATIBILITY	VARCHAR2(60)	Minimum software version required for a database instance to use files in this disk group

Note: The GROUP_NUMBER, TOTAL_MB, and FREE_MB columns are only meaningful if the disk group is mounted by the instance. Otherwise, their values will be 0.

V$ASM_DISKGROUP_STAT

V$ASM_DISKGROUP_STAT displays performance statistics in the same way that V$ASM_DISKGROUP does, but without performing discovery of new disk groups. This results in a less expensive operation. However, since discovery is not performed, the output of this view does not include any data about disk groups that are new to the system.

The columns for V$ASM_DISKGROUP_STAT are the same as those for V$ASM_DISKGROUP.

See Also: "V$ASM_DISKGROUP" on page 6-19

V$ASM_FILE

In an Automatic Storage Management instance, V$ASM_FILE displays one row for every Automatic Storage Management file in every disk group mounted by the Automatic Storage Management instance. In a database instance, V$ASM_FILE displays no rows.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Number of the disk group containing the file (composite primary key)
FILE_NUMBER	NUMBER	Number of the file within the disk group (composite primary key)
COMPOUND_INDEX	NUMBER	A 32-bit number consisting of a disk group number in the high-order 8 bits and a file number in the low-order 24 bits (for efficient access to the view)
INCARNATION	NUMBER	Incarnation number for the file (composite primary key)
BLOCK_SIZE	NUMBER	Block size of the file (in bytes)
BLOCKS	NUMBER	Number of blocks in the file

Column	Datatype	Description
BYTES	NUMBER	Number of bytes in the file
SPACE	NUMBER	Number of bytes allocated to the file
TYPE	VARCHAR2(64)	Type of the file. Possible values are as follows: ■ CONTROLFILE ■ DATAFILE ■ ONLINELOG ■ ARCHIVELOG ■ TEMPFILE ■ BACKUPSET ■ XTRANSPORT ■ PARAMETERFILE ■ DATAGUARDCONFIG ■ FLASHBACK ■ CHANGETRACKING ■ DUMPSET ■ AUTOBACKUP
REDUNDANCY	VARCHAR2(6)	Redundancy of the file: ■ UNPROT ■ MIRROR ■ PARITY ■ HIGH
STRIPED	VARCHAR2(6)	Indicates how the file is striped: ■ FINE ■ COARSE
CREATION_DATE	DATE	Date on which the file was created
MODIFICATION_DATE	DATE	Date of the last open/close for writing,rounded back to the nearest hour
REDUNDANCY_LOWERED	VARCHAR2(1)	Indicates whether a file has lower redundancy than what was expected (Y) or not (N). Redundancy is said to have been lowered for a file when one or more data extents in that file are not mirrored at the level specified by the administrator. In case of unprotected files, data extents could be missing altogether. Another possible value for this column is (U), which means that it is unknown.

V$ASM_OPERATION

In an Automatic Storage Management instance, V$ASM_OPERATION displays one row for every active Automatic Storage Management long running operation executing in the Automatic Storage Management instance. In a database instance, V$ASM_OPERATION displays no rows.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Disk group number (primary key). This is the foreign key to the V$ASM_DISKGROUP view.
OPERATION	CHAR(5)	Type of the operation: ■ REBAL - Rebalance pending for this group. The disk group needs rebalance to complete. ■ ONLIN - Online disks operation in progress for the disk group

Column	Datatype	Description
STATE	VARCHAR2(4)	State of the operation: - WAIT - No operations running for the group - RUN - Operation running for the group - REAP - Operation is being run down - HALT - Operation halted by admin - ERRS - Operation halted by errors
POWER	NUMBER	Power requested for the operation as specified by the ASM_POWER_LIMIT initialization parameter or command syntax
ACTUAL	NUMBER	Power allocated to the operation
SOFAR	NUMBER	Number of Allocation Units that have been moved so far by the operation
EST_WORK	NUMBER	Estimated number of Allocation Units that have to be moved by the operation
EST_RATE	NUMBER	Estimated number of Allocation Units that are being moved per minute by the operation
EST_MINUTES	NUMBER	Estimated amount of time (in minutes) that the remainder of the operation is expected to take
ERROR_CODE	VARCHAR2(44)	Oracle external error code; NULL if no error

V$ASM_TEMPLATE

In an Automatic Storage Management instance, V$ASM_TEMPLATE displays one row for every template present in every disk group mounted by the Automatic Storage Management instance. In a database instance, V$ASM_TEMPLATE displays one row for every template present in every disk group mounted by the Automatic Storage Management instance with which the database instance communicates.

Column	Datatype	Description
GROUP_NUMBER	NUMBER	Owning disk group number (foreign key to the V$ASM_DISKGROUP view)
ENTRY_NUMBER	NUMBER	Template number (primary key)
REDUNDANCY	VARCHAR2(6)	Redundancy of the template: - UNPROT - MIRROR - PARITY - HIGH
STRIPE	VARCHAR2(6)	Indicates how the template is striped: - FINE - COARSE
SYSTEM	VARCHAR2(1)	Indicates whether the template is a system template (Y) or not (N)
NAME	VARCHAR2(30)	Name of the template

V$AW_AGGREGATE_OP

V$AW_AGGREGATE_OP displays the aggregation operators available in analytic workspaces. You can use this view in an application to provide a list of choices.

Column	Datatype	Description
NAME	VARCHAR2(14)	Keyword for the aggregation operator
LONGNAME	VARCHAR2(30)	Descriptive name for the operator
DEFAULT_WEIGHT	NUMBER	Default weight factor for weighted operators

V$AW_ALLOCATE_OP

V$AW_ALLOCATE_OP displays the allocation operators available in analytic workspaces. You can use this view in an application to provide a list of choices.

Column	Datatype	Description
NAME	VARCHAR2(14)	Keyword for the allocation operator
LONGNAME	VARCHAR2(30)	Descriptive name for the operator

V$AW_CALC

V$AW_CALC reports on the effectiveness of various caches used by Oracle OLAP during dynamic aggregation.

Column	Datatype	Description
SESSION_ID	NUMBER	A unique numeric identifier for the session.
AGGREGATE_CACHE_HITS	NUMBER	The number of times a dimension member is found in the aggregate cache (a hit). The number of hits for run-time aggregation can be increased by fetching data across the dense dimension
AGGREGATE_CACHE_MISSES	NUMBER	The number of times a dimension member is not found in the aggregate cache and must be read from disk (a miss).
SESSION_CACHE_HITS	NUMBER	The number of times the data is found in the session cache (a hit).
SESSION_CACHE_MISSES	NUMBER	The number of times the data is not found in the session cache (a miss).
POOL_HITS	NUMBER	The number of times the data is found in a page in the OLAP page pool (a hit).
POOL_MISSES	NUMBER	The number of times the data is not found in the OLAP page pool (a miss).
POOL_NEW_PAGES	NUMBER	The number of newly created pages in the OLAP page pool that have not yet been written to the workspace LOB.
POOL_RECLAIMED_PAGES	NUMBER	The number of previously unused pages that have been recycled with new data.
CACHE_WRITES	NUMBER	The number of times the data from the OLAP page pool has been written to the database cache.
POOL_SIZE	NUMBER	The number of kilobytes in the OLAP page pool.
CURR_DML_COMMAND	VARCHAR2(64)	The command currently being executed.
PREV_DML_COMMAND	VARCHAR2(64)	The command most recently completed.
AGGR_FUNC_LOGICAL_NA	NUMBER	The number of times the aggregation engine returns a logical NA because the AGGINDEX option is on and the composite tuple does not exist.
AGGR_FUNC_PRECOMPUTE	NUMBER	The number of times the aggregation engine finds a value in a position that it was called to calculate.
AGGR_FUNC_CALCS	NUMBER	The number of times the aggregation engine calculates a parent value based on the values of its children.

V$AW_LONGOPS

V$AW_LONGOPS provides status information about active SQL cursors initiated in an analytic workspace.

Column	Datatype	Description
SESSION_ID	NUMBER	The identifier for the session in which the fetch is executing. This table can be joined with V$SESSION to obtain the user name.
CURSOR_NAME	VARCHAR2(64)	The name assigned to the cursor.

Column	Datatype	Description
COMMAND	VARCHAR2(7)	The command that is actively fetching data from relational tables.
STATUS	VARCHAR2(9)	One of the following values: ■ EXECUTING - the command has begun executing ■ FETCHING - data is being fetched into the analytic workspace ■ FINISHED - the command has finished executing. This status appears very briefly before the record disappears from the table.
ROWS_PROCESSED	NUMBER	The number of rows already inserted, updated, or deleted.
START_TIME	TIMESTAMP(3)	The time the command started executing.

V$AW_OLAP

V$AW_OLAP provides a record of active sessions and their use with analytic workspaces. A row is generated whenever an analytic workspace is created or attached. The first row for a session is created when the first command is issued. It identifies the SYS.EXPRESS workspace, which is attached automatically to each session. Rows related to a particular analytic workspace are deleted when the workspace is detached from the session or the session ends.

Column	Datatype	Description
SESSION_ID	NUMBER	A unique numeric identifier for a session
AW_NUMBER	NUMBER	A unique numeric identifier for an analytic workspace. To get the name of the analytic workspace, join this column to the AW_NUMBER column of the USER_AWS view or to the AWSEQ# column of the AW$ table.
ATTACH_MODE	VARCHAR2(10)	READ ONLY or READ WRITE
GENERATION	NUMBER	The generation of an analytic workspace. Each UPDATE creates a new generation. Sessions attaching the same workspace between UPDATE commands share the same generation.
TEMP_SPACE_PAGES	NUMBER	The number of pages stored in temporary segments for the analytic workspace.
TEMP_SPACE_READS	NUMBER	The number of times data has been read from a temporary segment and not from the page pool.
LOB_READS	NUMBER	The number of times data has been read from the table where the analytic workspace is stored (the permanent LOB).
POOL_CHANGED_PAGES	NUMBER	The number of pages in the page pool that have been modified in this analytic workspace.
POOL_UNCHANGED_PAGES	NUMBER	The number of pages in the page pool that have not been modified in this analytic workspace.

V$AW_SESSION_INFO

V$AW_SESSION_INFO provides information about each active session. A transaction is a single exchange between a client session and Oracle OLAP. Multiple commands can execute within a single transaction.

Column	Datatype	Description
SESSION_ID	NUMBER	A unique numeric identifier for a session
CLIENT_TYPE	VARCHAR2(64)	OLAP
SESSION_STATE	VARCHAR2(64)	TRANSACTING, NOT_TRANSACTING, EXCEPTION_HANDLING, CONSTRUCTING, CONSTRUCTED, DECONSTRUCTING, or DECONSTRUCTED
SESSION_HANDLE	NUMBER	The session identifier

Column	Datatype	Description
USERID	VARCHAR2(64)	The database user name under which the session opened
TOTAL_TRANSACTION	NUMBER	The total number of transactions executed within the session; this number provides a general indication of the level of activity in the session
TRANSACTION_TIME	NUMBER	The elapsed time in milliseconds of the mostly recently completed transaction
TOTAL_TRANSACTION_ TIME	NUMBER	The total elapsed time in milliseconds in which transactions were being executed
AVERAGE_TRANSACTION_ TIME	NUMBER	The average elapsed time in milliseconds to complete a transaction
TRANSACTION_CPU_TIME	NUMBER	The total CPU time in milliseconds used to complete the most recent transaction
TOTAL_TRANSACTION_ CPU_TIME	NUMBER	The total CPU time used to execute all transactions in this session; this total does not include transactions that are currently in progress
AVERAGE_TRANSACTION_ CPU_TIME	NUMBER	The average CPU time to complete a transaction; this average does not include transactions that are currently in progress

V$BACKUP

V$BACKUP displays the backup status of all online datafiles.

Column	Datatype	Description
FILE#	NUMBER	File identifier
STATUS	VARCHAR2(18)	File status: NOT ACTIVE, ACTIVE (backup in progress), OFFLINE NORMAL, or description of an error. NOT ACTIVE indicates that the file is not currently in backup mode (that is, an ALTER TABLESPACE ... BEGIN BACKUP or ALTER DATABASE BEGIN BACKUP statement has not been issued), whereas ACTIVE indicates that the file is currently in backup mode.
CHANGE#	NUMBER	System change number when backup started
TIME	DATE	Time the backup started

V$BACKUP_ARCHIVELOG_DETAILS

V$BACKUP_ARCHIVELOG_DETAILS contains information about all restorable archive logs. It will include all archived logs backed up in a backup set or proxy copies.

Column	Datatype	Description
BTYPE	CHAR(9)	Backup type container, either BACKUPSET or PROXYCOPY
BTYPE_KEY	NUMBER	Unique identifier for the backup type. For BACKUPSET, it is BS_KEY.
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
ID1	NUMBER	If BACKUPSET, it contains SET_STAMP If PROXYCOPY, it is RECID from the control file
ID2	NUMBER	If BACKUPSET, it contains SET_COUNT If PROXYCOPY, it is STAMP
THREAD#	NUMBER	Thread number
SEQUENCE#	NUMBER	Sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change SCN

Column	Datatype	Description
RESETLOGS_TIME	DATE	Resetlogs change time
FIRST_CHANGE#	NUMBER	First change SCN
FIRST_TIME	DATE	First change time
NEXT_CHANGE#	NUMBER	Next change SCN
NEXT_TIME	DATE	Next change time
FILESIZE	NUMBER	File size
COMPRESSION_RATIO	NUMBER	Compression ratio
FILESIZE_DISPLAY	VARCHAR2(4000)	Same value as the FILESIZE column, but converted to a user-displayable format, for example *n*M, *n*G, *n*T, *n*P, and so on

V$BACKUP_ARCHIVELOG_SUMMARY

V$BACKUP_ARCHIVELOG_SUMMARY provides archive log summary information based on archive logs in the backup set or on proxy copies.

Column	Datatype	Description
NUM_FILES_BACKED	NUMBER	Number of files backed up
NUM_DISTINCT_FILES_BACKED	NUMBER	Number of distinct archive log files backed up
MIN_FIRST_CHANGE#	NUMBER	Lowest SCN range value
MAX_NEXT_CHANGE#	NUMBER	Highest SCN range value
MIN_FIRST_TIME	DATE	Lowest SCN range time
MAX_NEXT_TIME	DATE	Highest SCH range time
INPUT_BYTES	NUMBER	Total input bytes read
OUTPUT_BYTES	NUMBER	Output size of backups
COMPRESSION_RATIO	NUMBER	Compression ratio
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for input bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$BACKUP_ASYNC_IO

V$BACKUP_ASYNC_IO displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

> **See Also:** *Oracle Database Performance Tuning Guide* for information on how to use this table to tune backup performance

Column	Datatype	Description
SID	NUMBER	Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	Use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
DEVICE_TYPE	VARCHAR2(17)	Device type where the file is located

Column	Datatype	Description
TYPE	VARCHAR2(9)	INPUT, OUTPUT, or AGGREGATE
STATUS	VARCHAR2(11)	NOT STARTED, IN PROGRESS, or FINISHED
FILENAME	VARCHAR2(513)	Name of the backup file being read or written
SET_COUNT	NUMBER	Set count of the backup set being read or written
SET_STAMP	NUMBER	Set stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	Size of the buffers being used to read/write the file, in bytes
BUFFER_COUNT	NUMBER	Number of buffers being used to read/write the file
TOTAL_BYTES	NUMBER	Total number of bytes that will be read or written for the file, if known. If not known, this column will be null
OPEN_TIME	DATE	Time the file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened
CLOSE_TIME	DATE	Time the file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed
ELAPSED_TIME	NUMBER	Time, in hundredths of a second, that the file was open
MAXOPENFILES	NUMBER	Number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'.
BYTES	NUMBER	Number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	Number of I/Os that were performed to this file
READY	NUMBER	Number of asynchronous requests for which a buffer was immediately ready for use
SHORT_WAITS	NUMBER	Number of times that a buffer was not immediately available, but a buffer became available after doing a nonblocking poll for I/O completion
SHORT_WAIT_TIME_TOTAL	NUMBER	Total time, in hundredths of a second, taken by nonblocking polls for I/O completion
SHORT_WAIT_TIME_MAX	NUMBER	Maximum time taken for a nonblocking poll for I/O completion, in hundredths of a second
LONG_WAITS	NUMBER	The number of times that a buffer was not immediately available, and only became available after a blocking wait was issued
LONG_WAIT_TIME_TOTAL	NUMBER	The total time, in hundredths of a second, taken by blocking waits for I/O completion
LONG_WAIT_TIME_MAX	NUMBER	The maximum time taken for a blocking wait for I/O completion, in hundredths of a second

V$BACKUP_CONTROLFILE_DETAILS

V$BACKUP_CONTROLFILE_DETAILS contains information about restorable control files. It will include all the control files backed up in the backup set, image copies, and proxy copies.

Column	Datatype	Description
BTYPE	CHAR(9)	Backup type container. Possible values are BACKUPSET, IMAGECOPY, PROXYCOPY.
BTYPE_KEY	NUMBER	Unique identifier for the backup type, either BS_KEY or COPY_KEY.
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp

Column	Datatype	Description
ID1	NUMBER	If BACKUPSET, it contains SET_STAMP If IMAGECOPY or PROXYCOPY, it is RECID from the control file
ID2	NUMBER	If BACKUPSET, it contains SET_COUNT If IMAGECOPY or PROXYCOPY, it is STAMP
CREATION_TIME	DATE	File creation time
RESETLOGS_CHANGE#	NUMBER	Resetlogs change SCN
RESETLOGS_TIME	DATE	Resetlogs change time
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change SCN
CHECKPOINT_TIME	DATE	Checkpoint change time
FILESIZE	NUMBER	File size, in bytes, for the output of backing up this control file
COMPRESSION_RATIO	NUMBER	Compression ratio for this backup
FILESIZE_DISPLAY	VARCHAR2(4000)	Same value as the FILESIZE column, but converted to a user-displayable format, for example *n*M, *n*G, *n*T, *n*P, and so on

V$BACKUP_CONTROLFILE_SUMMARY

V$BACKUP_CONTROLFILE_SUMMARY provides control file summary information, based on either a backup set of files, image copies, or proxy copies.

Column	Datatype	Description
NUM_FILES_BACKED	NUMBER	Number of files backed up for specific criteria
NUM_DISTINCT_FILES_BACKED	NUMBER	Number of distinct files backed up
MIN_CHECKPOINT_CHANGE#	NUMBER	Minimum checkpoint change number of the datafile for specified criteria
MAX_CHECKPOINT_CHANGE#	NUMBER	Maximum checkpoint change number of the datafile for specified criteria
MIN_CHECKPOINT_TIME	DATE	Minimum checkpoint time of the datafile for specified criteria
MAX_CHECKPOINT_TIME	DATE	Maximum checkpoint time of the datafile for specified criteria
INPUT_BYTES	NUMBER	Total input bytes of files read
OUTPUT_BYTES	NUMBER	Total output bytes written
COMPRESSION_RATIO	NUMBER	Compression ratio
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for input bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$BACKUP_COPY_DETAILS

V$BACKUP_COPY_DETAILS contains information about all available control file and datafile copies.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
COPY_KEY	NUMBER	Unique identifier for this datafile or control file copy
FILE#	NUMBER	Absolute datafile number

Column	Datatype	Description
NAME	VARCHAR2(513)	File name of the datafile copy. The maximum length of the name is dependent on your operating system.
TAG	VARCHAR2(32)	Datafile copy tag
CREATION_CHANGE#	NUMBER	Datafile creation change number
CREATION_TIME	DATE	Datafile creation timestamp
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt by this copy operation. That is, blocks that were not marked corrupted in the source datafile, but were detected and marked as corrupted during the copy operation.
OUTPUT_BYTES	NUMBER	Total output bytes written
COMPLETION_TIME	DATE	Time when the copy was completed
CONTROLFILE_TYPE	VARCHAR2(1)	Type of control file. B indicates normal copies. S indicates standby copies.
KEEP	VARCHAR2(3)	(YES|NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy
KEEP_UNTIL	DATE	IF specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(10)	Lists additional retention options for this backup set. Possible values are: ■ LOGS - The logs needed to recover this backup set are kept ■ NOLOGS - The logs needed to recover this backup set are not kept
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether or not the file was created in the flash recovery area (YES) or not (NO)
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Size of backup set to display

V$BACKUP_COPY_SUMMARY

V$BACKUP_COPY_SUMMARY provides summary information for the output datafile and control file copy.

Column	Datatype	Description
NUM_COPIES	NUMBER	Number of copies created
NUM_DISTINCT_COPIES	NUMBER	Number of distinct copies (that contain datafiles with different checkpoints)
MIN_CHECKPOINT_CHANGE#	NUMBER	Minimum checkpoint change SCN
MAX_CHECKPOINT_CHANGE#	NUMBER	Maximum checkpoint change SCN
MIN_CHECKPOINT_TIME	DATE	Minimum checkpoint change time
MAX_CHECKPOINT_TIME	DATE	Maximum checkpoint change time
OUTPUT_BYTES	NUMBER	Total number of output bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$BACKUP_CORRUPTION

V$BACKUP_CORRUPTION displays information about corrupt block ranges in datafile backups from the control file. Note that corruptions are not tolerated in the control file and archived redo log backups.

Column	Datatype	Description
RECID	NUMBER	Backup corruption record ID
STAMP	NUMBER	Backup corruption record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	backup piece that contains this corrupt block
FILE#	NUMBER	Absolute file number of the datafile that contains the corrupt blocks
BLOCK#	NUMBER	Block number of the first corrupt block in the range of corrupted blocks
BLOCKS	NUMBER	Number of corrupted blocks found starting with BLOCK#
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
MARKED_CORRUPT	VARCHAR2(3)	Indicates whether this corruption was not previously detected by the Oracle Database (YES) or the Oracle Database had already discovered this corrupt block and marked it as corrupt (NO). Note that when a corrupt block is encountered in a backup, and was not already marked corrupt by the Oracle Database, then the backup process does not mark the block as corrupt in the production datafile. Thus, this field may be YES for the same block in more than one backup set.
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: ▪ ALL ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ▪ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ▪ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ▪ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ▪ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V$BACKUP_DATAFILE

V$BACKUP_DATAFILE displays information about control files and datafiles in backup sets from the control file.

Column	Datatype	Description
RECID	NUMBER	Backup datafile record ID
STAMP	NUMBER	Backup datafile record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
FILE#	NUMBER	Datafile number; set to 0 for control file
CREATION_CHANGE#	NUMBER	Creation system change number (SCN) of the datafile
CREATION_TIME	DATE	Creation timestamp of the datafile
RESETLOGS_CHANGE#	NUMBER	Resetlogs system change number (SCN) of the datafile when it was backed up
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when it was backed up
INCREMENTAL_LEVEL	NUMBER	(0-4) incremental backup level
INCREMENTAL_CHANGE#	NUMBER	All blocks changed after the incremental change number is included in this backup; set to 0 for a full backup

Column	Datatype	Description
CHECKPOINT_CHANGE#	NUMBER	All changes up to the checkpoint change number are included in this backup
CHECKPOINT_TIME	DATE	Timestamp of the checkpoint
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change number in this backup
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt
MEDIA_CORRUPT	NUMBER	Number of blocks media corrupt
LOGICALLY_CORRUPT	NUMBER	Number of blocks logically corrupt
DATAFILE_BLOCKS	NUMBER	Size of the datafile in blocks at backup time. This value is also the number of blocks taken by the datafile restarted from this backup.
BLOCKS	NUMBER	Size of the backup datafile (in blocks). Unused blocks are not copied to the backup.
BLOCK_SIZE	NUMBER	Block size
OLDEST_OFFLINE_RANGE	NUMBER	RECID of the oldest offline range record in this backup control file. 0 for datafile backups.
COMPLETION_TIME	DATE	Time completed
CONTROLFILE_TYPE	VARCHAR2(1)	B - Normal copies S - Standby copies
USED_CHANGE_TRACKING	VARCHAR2(3)	Indicates whether change tracking data was used to accelerate this incremental backup (YES) or whether change tracking data was not used (NO)
BLOCKS_READ	NUMBER	Number of blocks that were scanned while taking this backup. If this was an incremental backup, and change tracking was used to optimize the backup, then the value of this column will be smaller than DATAFILE_BLOCKS. Otherwise, the value of this column will be the same as DATAFILE_BLOCKS. Even when change tracking data is used, the value of this column may be larger than BLOCKS, because the data read by change tracking is further refined during the process of creating an incremental backup.
USED_OPTIMIZATION	VARCHAR2(3)	Whether or not backup optimization was applied
FOREIGN_DBID	NUMBER	Foreign DBID of the database from which this datafile was transported. The value is 0 if the file backed up is not a foreign database file.
PLUGGED_READONLY	VARCHAR2(3)	YES if this is a backup of a transported read-only foreign file; otherwise NO.
PLUGIN_CHANGE#	NUMBER	SCN at which the foreign datafile was transported into the database. The value is 0 if this file is not a foreign database file.
PLUGIN_RESETLOGS_CHANGE#	NUMBER	The SCN of the RESETLOGS operation for the incarnation into which this foreign file was transported. The value is 0 if this file is not a foreign database file.
PLUGIN_RESETLOGS_TIME	DATE	The time of the RESETLOGS operation for the incarnation into which this foreign file was transported. The value is 0 if this file is not a foreign database file.
SECTION_SIZE	NUMBER	Specifies the number of blocks in each section of a multisection backup. Value is 0 for whole file backups.

V$BACKUP_DATAFILE_DETAILS

V$BACKUP_DATAFILE_DETAILS contains information about restorable datafiles. It will include all datafiles backed in the backup set, image copies, and proxy copies.

Column	Datatype	Description
BTYPE	CHAR(9)	Backup type container. Possible values are: BACKUPSET, IMAGECOPY, PROXYCOPY.

Column	Datatype	Description
BTYPE_KEY	NUMBER	Unique identifier for the backup type. For BACKUPSET, it is BS_KEY.
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session record ID
SESSION_STAMP	NUMBER	Session stamp
ID1	NUMBER	If BACKUPSET, it contains SET_STAMP. If IMAGECOPY or PROXYCOPY, it is RECID from the control file.
ID2	NUMBER	If BACKUPSET, it contains SET_COUNT. If IMAGECOPY or PROXYCOPY, it is STAMP.
FILE#	NUMBER	File number
CREATION_CHANGE#	NUMBER	File creation change SCN
CREATION_TIME	DATE	File creation time
RESETLOGS_CHANGE#	NUMBER	Resetlogs change SCN
RESETLOGS_TIME	DATE	Resetlogs change time
INCREMENTAL_LEVEL	NUMBER	Incremental level
INCREMENTAL_CHANGE#	NUMBER	Incremental change SCN
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change SCN
CHECKPOINT_TIME	DATE	Checkpoint change time
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt
FILESIZE	NUMBER	File size, in bytes
COMPRESSION_RATIO	NUMBER	Compression ratio
TS#	NUMBER	Tablespace number
TSNAME	VARCHAR2(30)	Tablespace name
FILESIZE_DISPLAY	VARCHAR2(4000)	Same value as the FILESIZE column, but converted to a user-displayable format, for example *n*M, *n*G, *n*T, *n*P, and so on

V$BACKUP_DATAFILE_SUMMARY

V$BACKUP_DATAFILE_SUMMARY provides summary information for a specific criteria set, based on a backup job, a time range applicable to jobs, or a specific datafile).

Column	Datatype	Description
NUM_FILES_BACKED	NUMBER	Number of files backed up for specified criteria
NUM_DISTINCT_FILES_BACKED	NUMBER	Number of distinct files backed up
NUM_DISTINCT_TS_BACKED	NUMBER	Number of distinct tablespaces backed up
MIN_CHECKPOINT_CHANGE#	NUMBER	Minimum checkpoint change number of the datafile for specified criteria
MAX_CHECKPOINT_CHANGE#	NUMBER	Maximum checkpoint change number of the datafile for specified criteria
MIN_CHECKPOINT_TIME	DATE	Minimum checkpoint time of the datafile for specified criteria
MAX_CHECKPOINT_TIME	DATE	Maximum checkpoint time of the datafile for specified criteria
INPUT_BYTES	NUMBER	Total input bytes of files read
OUTPUT_BYTES	NUMBER	Total output bytes written
COMPRESSION_RATIO	NUMBER	Compression ratio
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for input bytes

Column	Datatype	Description
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$BACKUP_DEVICE

V$BACKUP_DEVICE displays information about supported backup devices. If a device type does not support named devices, then one row with the device type and a null device name is returned for that device type. If a device type supports named devices then one row is returned for each available device of that type. The special device type DISK is not returned by this view because it is always available.

Column	Datatype	Description
DEVICE_TYPE	VARCHAR2(17)	Type of the backup device
DEVICE_NAME	VARCHAR2(513)	Name of the backup device

V$BACKUP_FILES

V$BACKUP_FILES displays information about all RMAN backups (both image copies and backup sets) and archived logs.

This view simulates the LIST BACKUP and LIST COPY RMAN commands. This view requires that the database be set using the DBMS_RCVMAN.SETDATABASE procedure.

Column	Datatype	Description
PKEY	NUMBER	Primary key for the backup
BACKUP_TYPE	VARCHAR2(32)	Type of the backup: ■ BACKUP SET ■ COPY ■ PROXY COPY
FILE_TYPE	VARCHAR2(32)	Type of the file: ■ DATAFILE ■ CONTROLFILE ■ SPFILE ■ REDO LOG ■ COPY (for an image copy backup) ■ PIECE (for a backup piece)
KEEP	VARCHAR2(3)	Indicates whether the backup has a retention policy different from the value for CONFIGURE RETENTION POLICY (YES) or not (NO)
KEEP_UNTIL	DATE	If the KEEP UNTIL TIME clause of the BACKUP command was specified, then this column shows the date after which the backup becomes obsolete. If the column is null and KEEP_OPTIONS is not null, the backup never becomes obsolete.
KEEP_OPTIONS	VARCHAR2(13)	KEEP options for the backup: ■ LOGS - RMAN keeps the logs needed to recover the backup ■ NOLOGS - RMAN does not keep the logs needed to recover the backup If this column is null, then the backup has no KEEP options and will be made obsolete based on the retention policy.

Column	Datatype	Description
STATUS	VARCHAR2(16)	Status of the backup: ■ AVAILABLE ■ UNAVAILABLE ■ EXPIRED ■ OTHER
FNAME	VARCHAR2(1024)	Name of the file
TAG	VARCHAR2(32)	Tag of the piece, copy, or proxy copy
MEDIA	VARCHAR2(80)	Media ID of the piece or proxy copy
RECID	NUMBER	Recid of the record in the control file
STAMP	NUMBER	Stamp of the record in the control file
DEVICE_TYPE	VARCHAR2(255)	Type of media device that stores the backup
BLOCK_SIZE	NUMBER	Block size for the backup (in bytes)
COMPLETION_TIME	DATE	Time when the backup completed
COMPRESSED	VARCHAR2(3)	Indicates whether the backup piece is compressed (YES) or not (NO); valid only if FILE_TYPE is PIECE. Image copies cannot be compressed.
OBSOLETE	VARCHAR2(3)	Indicates whether the backup piece or copy is obsolete (YES) or not (NO); valid only if FILE_TYPE is PIECE or COPY
BYTES	NUMBER	Size of the file (in bytes)
BS_KEY	NUMBER	Primary key of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_COUNT	NUMBER	Count of the backup set from the control file record (valid only if BACKUP_TYPE is BACKUP SET)
BS_STAMP	NUMBER	Stamp of the backup set from the control file record (valid only if BACKUP_TYPE is BACKUP SET)
BS_TYPE	VARCHAR2(32)	Type of the backup set (valid only if BACKUP_TYPE is BACKUP SET): ■ DATAFILE ■ ARCHIVED LOG
BS_INCR_TYPE	VARCHAR2(32)	Incremental type of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_PIECES	NUMBER	Number of backup pieces in the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_COPIES	NUMBER	Number of copies of the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BS_COMPLETION_TIME	DATE	Completion time of the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_STATUS	VARCHAR2(16)	Status of the backup set (valid only if BACKUP_TYPE is BACKUP SET): ■ AVAILABLE ■ UNAVAILABLE ■ EXPIRED ■ OTHER - Pieces which are part of the backup set do not have uniform status (that is, some of them are available, some not)
BS_BYTES	NUMBER	Sum of all backup piece sizes in the backup set (valid only if BACKUP_TYPE is BACKUP SET)
BS_COMPRESSED	VARCHAR2(3)	Indicates whether the backup pieces of the backup set are compressed (YES) or not (NO); valid only if BACKUP_TYPE is BACKUP SET
BS_TAG	VARCHAR2(1024)	Tags of the backup set. If pieces have different tags, then all piece tags are concatenated and separated by commas. Valid only if BACKUP_TYPE is BACKUP SET)
BS_DEVICE_TYPE	VARCHAR2(255)	Device type of the backup set. If there is more than one device type, then they are separated by commas. Valid only if BACKUP_TYPE is BACKUP SET

Column	Datatype	Description
BP_PIECE#	NUMBER	Number of pieces inside the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BP_COPY#	NUMBER	Number of copies of the backup set (valid only if FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
DF_FILE#	NUMBER	Absolute file number of the datafile (valid only if FILE_TYPE is DATAFILE)
DF_TABLESPACE	VARCHAR2(30)	Tablespace name of the datafile (valid only if FILE_TYPE is DATAFILE)
DF_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the control file or datafile was created (valid only if FILE_TYPE is DATAFILE)
DF_CREATION_CHANGE#	NUMBER	Creation SCN of the control file or datafile (valid only if FILE_TYPE is DATAFILE)
DF_CHECKPOINT_CHANGE#	NUMBER	System change number (SCN) of the most recent control file or datafile checkpoint (valid only if FILE_TYPE is DATAFILE)
DF_CKP_MOD_TIME	DATE	Modification time in case of SPFILE, otherwise time when the control file or datafile was checkpointed (valid only if FILE_TYPE is DATAFILE)
RL_THREAD#	NUMBER	Redo log thread number of the archived log (valid only if FILE_TYPE is REDO LOG)
RL_SEQUENCE#	NUMBER	Redo log sequence number of the archived log (valid only if FILE_TYPE is REDO LOG)
RL_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the record was created (valid only if FILE_TYPE is REDO LOG)
RL_FIRST_CHANGE#	NUMBER	First SCN of the redo log (valid only if FILE_TYPE is REDO LOG)
RL_FIRST_TIME	DATE	Time when the Oracle Database switched into the redo log (valid only if FILE_TYPE is REDO LOG)
RL_NEXT_CHANGE#	NUMBER	First SCN of the next redo log in the thread (valid only if FILE_TYPE is REDO LOG)
RL_NEXT_TIME	DATE	First timestamp of the next redo log in the thread (valid only if FILE_TYPE is REDO LOG)

V$BACKUP_PIECE

V$BACKUP_PIECE displays information about backup pieces from the control file. Each backup set consists of one or more backup pieces.

Column	Datatype	Description
RECID	NUMBER	Backup piece record ID
STAMP	NUMBER	Backup piece record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	Backup piece number (1-N)
COPY#	NUMBER	Indicates the copy number for backup pieces created with duplex enabled. 1 if the backup piece is not duplexed.
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the backup piece resides. Set to DISK for backup sets on disk. **See Also:** V$BACKUP_DEVICE
HANDLE	VARCHAR2(513)	Backup piece handle identifies the backup piece on restore
COMMENTS	VARCHAR2(64)	Comment returned by the operating system or storage subsystem. Set to NULL for backup pieces on disk. This value is informational only; not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the backup piece resides. This value is informational only; not needed for restore.

Column	Datatype	Description
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.
CONCUR	VARCHAR2(3)	(YES | NO) Indicates whether the piece on a media that can be accessed concurrently
TAG	VARCHAR2(32)	Backup piece tag. The tag is specified at backup set level, but stored at piece level.
STATUS	VARCHAR2(1)	Indicates the status of the piece: A (available), D (deleted), or X (expired)
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
DELETED	VARCHAR2(3)	(YES/NO) NO indicates that the file still exists. YES indicates the file no longer exists because it has been deleted.
BYTES	NUMBER	Size of the backup piece (in bytes)
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
COMPRESSED	VARCHAR2(3)	Indicates whether the backup piece is compressed (YES) or not (NO)
BACKED_BY_VSS	VARCHAR2(3)	Whether or not the file has been backed up by Volume Shadow Copy Service (VSS). This column is for internal use only.

V$BACKUP_PIECE_DETAILS

V$BACKUP_PIECE_DETAILS displays information about all available backup pieces.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
BS_KEY	NUMBER	Backup set identifier
BP_KEY	NUMBER	Backup piece key
RECID	NUMBER	Backup piece record ID
STAMP	NUMBER	Backup piece record stamp
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count
PIECE#	NUMBER	Backup piece number (1-N)
COPY#	NUMBER	Indicates the copy number for backup pieces created with duplex enabled. The value is 1 if the backup piece is not duplexed.
DEVICE_TYPE	VARCHAR2(17)	Type of device on which the backup piece resides. Set to DISK for backup sets on disk.
HANDLE	VARCHAR2(513)	Backup piece handle identifies the backup piece on restore
COMMENTS	VARCHAR2(64)	Comments returned by the operating system or storage subsystem. Set to NULL for backup pieces on disk. This value is informational only. It is not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the backup piece resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	The media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.

Column	Datatype	Description
CONCUR	VARCHAR2(3)	(YES|NO) indicates whether or not the piece is on a media that can be accessed concurrently
TAG	VARCHAR2(32)	Backup piece tag. The tag is specified at backup set level, but stored at piece level.
STATUS	VARCHAR2(1)	Indicates the status of the piece: A (available), D (deleted), or X (expired)
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
DELETED	VARCHAR2(3)	NO indicates that the file still exists. YES indicates that the file no longer exists because it has been deleted.
BYTES	NUMBER	Size of the backup piece, in bytes
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether or not the file was created in the flash recovery area (YES) or not (NO)
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
COMPRESSED	VARCHAR2(3)	Indicates whether the backup piece is compressed (YES) or not (NO)
BACKED_BY_VSS	VARCHAR2(3)	Whether or not the file has been backed up by Volume Shadow Copy Service (VSS). This column is for internal use only.
PIECES_PER_SET	NUMBER	Number of backup pieces per set
SIZE_BYTES_DISPLAY	VARCHAR2(4000)	Size (in bytes) of the backup piece to display

V$BACKUP_REDOLOG

V$BACKUP_REDOLOG displays information about archived logs in backup sets from the control file. Note that online redo logs cannot be backed up directly; they must be archived first to disk and then backed up. An archive log backup set can contain one or more archived logs.

Column	Datatype	Description
RECID	NUMBER	Record ID for this row; it is an integer that identifies this row
STAMP	NUMBER	Timestamp used with RECID to uniquely identify this row
SET_STAMP	NUMBER	One of the foreign keys for the row of the V$BACKUP_SET table that identifies this backup set
SET_COUNT	NUMBER	One of the foreign keys for the row of the V$BACKUP_SET table that identifies this backup set
THREAD#	NUMBER	Thread number for the log
SEQUENCE#	NUMBER	Log sequence number
RESETLOGS_CHANGE#	NUMBER	Change number of the last resetlogs before the log was written
RESETLOGS_TIME	DATE	Change time of the last resetlogs before the log was written. These will be the same for all logs in a backup set.
FIRST_CHANGE#	NUMBER	SCN when the log was switched into. The redo in the log is at this SCN and greater.
FIRST_TIME	DATE	Time allocated when the log was switched into
NEXT_CHANGE#	NUMBER	SCN when the next log in this thread was switched into. The redo in the log is below this SCN.
NEXT_TIME	DATE	Time when the next log in this thread was switched into
BLOCKS	NUMBER	Size of the log in logical blocks including the header block
BLOCK_SIZE	NUMBER	Size of the log blocks in bytes

Column	Datatype	Description
TERMINAL	VARCHAR2(3)	???

V$BACKUP_SET

V$BACKUP_SET displays information about backup sets from the control file. A backup set record is inserted after the backup set is successfully completed.

Column	Datatype	Description
RECID	NUMBER	Backup set record ID
STAMP	NUMBER	Backup set record stamp
SET_STAMP	NUMBER	Backup set stamp. The backup set stamp and count uniquely identify the backup set. Primary key for the V$BACKUP_SET table, and the foreign key for the following tables: V$BACKUP_PIECE, V$BACKUP_DATAFILE, V$BACKUP_REDOLOG, V$BACKUP_CORRUPTION.
SET_COUNT	NUMBER	Backup set count. The backup set count is incremented by one every time a new backup set is started (if the backup set is never completed the number is "lost"). If the control file is re-created then the count is reset to 1. Therefore the count must be used with the stamp to uniquely identify a backup set. Primary key for the V$BACKUP_SET table, and the foreign key for the following tables: V$BACKUP_PIECE, V$BACKUP_DATAFILE, V$BACKUP_REDOLOG, V$BACKUP_CORRUPTION
BACKUP_TYPE	VARCHAR2(1)	Type of files that are in this backup. If the backup contains archived redo logs, the value is L. If this is a datafile full backup, the value is D. If this is an incremental backup, the value is I.
CONTROLFILE_INCLUDED	VARCHAR2(3)	Set to YES if there is a control file included in this backup set, otherwise set to NO
INCREMENTAL_LEVEL	NUMBER	Location where this backup set fits into the database's backup strategy. Set to zero for full datafile backups, nonzero for incremental datafile backups, and NULL for archivelog backups.
PIECES	NUMBER	Number of distinct backup pieces in the backup set
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Time that this backup set completed
ELAPSED_SECONDS	NUMBER	The number of elapsed seconds
BLOCK_SIZE	NUMBER	Block size of the backup set
INPUT_FILE_SCAN_ONLY	VARCHAR2(3)	YES indicates no actual backup is performed, but the datafiles are read. NO indicates a normal backup is performed.
KEEP	VARCHAR2(3)	(YES/NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy
KEEP_UNTIL	DATE	If KEEP_UNTIL_TIME is specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(10)	Lists additional retention options for this backup set. Possible values are: LOGS - The logs need to recover this backup are kept NOLOGS - The logs needed to recover this backup will not be kept BACKUP_LOGS - An archive log backup exists to support this backup set
MULTI_SECTION	VARCHAR2(3)	Indicates whether or not this backup set is a multi-section backup. Valid values are YES and NO. A multi-section backup is a backup in which multiple backup pieces are produced independently in parallel by multiple channels.

V$BACKUP_SET_DETAILS

V$BACKUP_SET_DETAILS provides detailed information about the backup set. This view will contain an extra row for each backup session that invokes BACKUP BACKUPSET (that is, creates new copies for the same backup set or copies backup set information from disk to tape). However, the remaining values of other columns belong to the complete backup set.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Identifies the job, together with SESSION_STAMP.
SESSION_STAMP	NUMBER	Identifies the job, together with SESSION_RECID
BS_KEY	NUMBER	Backup set identifier
RECID	NUMBER	RECID from V$BACKUP_SET record
STAMP	NUMBER	Stamp from V$BACKUP_SET record
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup count number
BACKUP_TYPE	VARCHAR2(1)	Type of backup (same as in V$BACKUP_SET)
CONTROLFILE_INCLUDED	VARCHAR2(3)	Control file included in backup set (same as in V$BACKUP_SET)
INCREMENTAL_LEVEL	NUMBER	Incremental level (same as in V$BACKUP_SET)
PIECES	NUMBER	Number of pieces (same as in V$BACKUP_SET)
START_TIME	DATE	Start time of the backup set (same as in V$BACKUP_SET)
COMPLETION_TIME	DATE	Completion time of the backup set (same as in V$BACKUP_SET)
ELAPSED_SECONDS	NUMBER	Time taken for backup set creation (same as in V$BACKUP_SET)
BLOCK_SIZE	NUMBER	Block size (same as in V$BACKUP_SET)
KEEP	VARCHAR2(3)	Keep value (same as in V$BACKUP_SET)
KEEP_UNTIL	DATE	Keep Until time (same as in V$BACKUP_SET)
KEEP_OPTIONS	VARCHAR2(10)	Keep options (same as in V$BACKUP_SET)
DEVICE_TYPE	VARCHAR2(17)	Type of device. If the backup set exists on more than one device type, an * is indicated here.
COMPRESSED	VARCHAR2(3)	YES, if backup is compressed
NUM_COPIES	NUMBER	Number of identical copies
OUTPUT_BYTES	NUMBER	Size of the backup set
ORIGINAL_INPUT_BYTES	NUMBER	Amount of data backed up when the backup set was created
COMPRESSION_RATIO	NUMBER	Compression ratio (input size/output size)
STATUS	CHAR(1)	The status of the backup set. It is always A (all backup pieces available), because this view only reflects available backup sets.
ORIGINAL_INPRATE_BYTES	NUMBER	Number of bytes read per second when backup set was initially created
OUTPUT_RATE_BYTES	NUMBER	Number of bytes written per second when the backup set was initially created
ORIGINAL_INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Input rate to display
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Size of backup set to display
ORIGINAL_INPRATE_BYTES_DISPLAY	VARCHAR2(4000)	Input rate per second for display
OUTPUT_RATE_BYTES_DISPLAY	VARCHAR2(4000)	Output rate per second for display
TIME_TAKEN_DISPLAY	VARCHAR2(4000)	Elapsed time in hh:mm:ss format

V$BACKUP_SET_SUMMARY

V$BACKUP_SET_SUMMARY provides summary information for a backup set.

Column	Datatype	Description
NUM_BACKUPSETS	NUMBER	Total number of backup sets created
OLDEST_BACKUP_TIME	DATE	Oldest backup start time
NEWEST_BACKUP_TIME	DATE	Newest backup start time
OUTPUT_BYTES	NUMBER	Number of output bytes (not including multiple copies)
ORIGINAL_INPUT_BYTES	NUMBER	Number of input bytes when backup sets were created
ORIGINAL_INPRATE_BYTES	NUMBER	Average input rate
OUTPUT_RATE_BYTES	NUMBER	Average output rate
COMPRESSION_RATIO	NUMBER	Compression ratio
ORIGINAL_INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for input bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes
ORIGINAL_INPRATE_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for input rate
OUTPUT_RATE_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output rate

V$BACKUP_SPFILE

V$BACKUP_SPFILE displays information about server parameter files in backup sets from the control file.

Column	Datatype	Description
RECID	NUMBER	Backup SPFILE record ID
STAMP	NUMBER	Backup SPFILE record stamp
SET_STAMP	NUMBER	Backup set stamp (of the set which contains this SPFILE backup)
SET_COUNT	NUMBER	Backup set count (of the set which contains this SPFILE backup)
MODIFICATION_TIME	DATE	Time when the SPFILE was last modified (this also includes creation time)
BYTES	NUMBER	Size of the SPFILE (in bytes)
COMPLETION_TIME	DATE	Time when the backup of the SPFILE completed
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name

V$BACKUP_SPFILE_DETAILS

V$BACKUP_SPFILE_DETAILS displays information about all restorable SP files backed up in the backup set.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
BS_KEY	NUMBER	Unique backup set identifier
SET_STAMP	NUMBER	Backup set stamp
SET_COUNT	NUMBER	Backup set count

Column	Datatype	Description
MODIFICATION_TIME	DATE	Time the backup set was modified
FILESIZE	NUMBER	Size, in bytes, of the SPFILE that was backed up
FILESIZE_DISPLAY	VARCHAR2(4000)	Same value as the FILESIZE column, but converted to a user-displayable format, for example *n*M, *n*G, *n*T, *n*P, and so on

V$BACKUP_SPFILE_SUMMARY

V$BACKUP_SPFILE_SUMMARY provides summary information for input SP file, based on either a backup job or time range applicable to jobs.

Column	Datatype	Description
NUM_FILES_BACKED	NUMBER	Number of files backed up
NUM_DISTINCT_FILES_BACKED	NUMBER	Number of distinct SP files backed up (with modification timestamp)
MIN_MODIFICATION_TIME	DATE	Minimum modification time
MAX_MODIFICATION_TIME	DATE	Maximum modification time
INPUT_BYTES	NUMBER	Total input bytes for all SP files backed up
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for all input bytes

V$BACKUP_SYNC_IO

V$BACKUP_SYNC_IO displays performance information about ongoing and recently completed RMAN backups and restores. For each backup, it contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not stored persistently, and is not preserved when the instance is re-started.

> **See Also:** *Oracle Database Performance Tuning Guide* for information on how to use this table to tune backup performance

Column	Datatype	Description
SID	NUMBER	The Oracle SID of the session doing the backup or restore
SERIAL	NUMBER	The use count for the SID doing the backup or restore
USE_COUNT	NUMBER	A counter that can be used to identify rows from different backup sets
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
DEVICE_TYPE	VARCHAR2(17)	The device type where the file is located
TYPE	VARCHAR2(9)	INPUT, OUTPUT, or AGGREGATE
STATUS	VARCHAR2(11)	NOT STARTED, IN PROGRESS, or FINISHED
FILENAME	VARCHAR2(513)	The name of the backup file being read or written
SET_COUNT	NUMBER	The set count of the backup set being read or written
SET_STAMP	NUMBER	The set stamp of the backup set being read or written
BUFFER_SIZE	NUMBER	The size of the buffers being used to read/write this file, in bytes
BUFFER_COUNT	NUMBER	The number of buffers being used to read/write this file
TOTAL_BYTES	NUMBER	The total number of bytes that will be read or written for this file, if known. If not known, this column will be null.

Column	Datatype	Description
OPEN_TIME	DATE	The time this file was opened. If TYPE='AGGREGATE', then this is the time that the first file in the aggregate was opened.
CLOSE_TIME	DATE	The time this file was closed. If TYPE='AGGREGATE', then this is the time that the last file in the aggregate was closed.
ELAPSED_TIME	NUMBER	The time, in hundredths of a second, that the file was open
MAXOPENFILES	NUMBER	The number of concurrently open DISK files. This value is only present in rows where TYPE='AGGREGATE'.
BYTES	NUMBER	The number of bytes read or written so far
EFFECTIVE_BYTES_PER_SECOND	NUMBER	The I/O rate that was achieved with this device during this backup
IO_COUNT	NUMBER	The number of I/Os that were performed to this file
IO_TIME_TOTAL	NUMBER	The total time, in hundredths of a second, taken to do I/O for this file
IO_TIME_MAX	NUMBER	The maximum time taken for a single I/O request
DISCRETE_BYTES_PER_SECOND	NUMBER	The average transfer rate for this file

V$BGPROCESS

V$BGPROCESS displays information about the background processes.

Column	Datatype	Description
PADDR	RAW(4 | 8)	Address of the process state object
PSERIAL#	NUMBER	Process state object serial number
NAME	VARCHAR2(5)	Name of this background process
DESCRIPTION	VARCHAR2(64)	Description of the background process
ERROR	NUMBER	Error encountered

V$BH

V$BH displays the status and number of pings for every buffer in the SGA. This is a Real Application Clusters view.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V$DBFILE)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(6)	Status of the buffer: ▪ free - Not currently in use ▪ xcur - Exclusive ▪ scur - Shared current ▪ cr - Consistent read ▪ read - Being read from disk ▪ mrec - In media recovery mode ▪ irec - In instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.

Column	Datatype	Description
LOCK_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_CLASS	NUMBER	The address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
DIRTY	VARCHAR2(1)	Y - block modified
TEMP	VARCHAR2(1)	Y - temporary block
PING	VARCHAR2(1)	Y - block pinged
STALE	VARCHAR2(1)	Y - block is stale
DIRECT	VARCHAR2(1)	Y - direct block
NEW	VARCHAR2(1)	Always set to N. This column is obsolete and maintained for backward compatibility.
OBJD	NUMBER	Database object number of the block that the buffer represents
TS#	NUMBER	Tablespace number of block

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

V$BLOCK_CHANGE_TRACKING

V$BLOCK_CHANGE_TRACKING displays the status of block change tracking for the database.

Column	Datatype	Description
STATUS	VARCHAR2(10)	Status of block change tracking in the database: ■ DISABLED - Block change tracking is disabled ■ TRANSITION - Block change tracking is in the process of transitioning between the enabled and disabled states. The TRANSITION state should usually never be observed, because it only exists while enabling or disabling block change tracking. This state might be observed if the instance crashed while enabling or disabling block change tracking, in which case it will be cleaned up automatically the next time that the database is opened. ■ ENABLED - Block change tracking is enabled
FILENAME	VARCHAR2(513)	Name of the block change tracking file for the database
BYTES	NUMBER	Size of the block change tracking file (in bytes)

See Also: *Oracle Database Backup and Recovery Basics* for information on setting up block change tracking

V$BLOCKING_QUIESCE

V$BLOCKING_QUIESCE indicates if a session is blocking, or would block, a quiesce operation.

Column	Datatype	Description
SID	NUMBER	Session identifier

V$BUFFER_POOL

V$BUFFER_POOL displays information about all buffer pools available for the instance.

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier number
NAME	VARCHAR2(20)	Name of the buffer pool: ■ DEFAULT ■ KEEP ■ RECYCLE **Note:** Currently, KEEP and RECYCLE pools only exist for the standard block size. All non-standard block size pools are DEFAULT.
BLOCK_SIZE	NUMBER	Block size (in bytes) for buffers in this pool. Possible values: the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, 32768.
RESIZE_STATE	VARCHAR2(10)	Current state of the resize operation: STATIC - Not being resized ALLOCATING - Memory is being allocated (can be cancelled by the user) ACTIVATING - New buffers are being created (user cannot cancel) SHRINKING - Buffers are being deleted (can be cancelled by the user)
CURRENT_SIZE	NUMBER	Present size of the sub-cache (in megabytes)
BUFFERS	NUMBER	Current instantaneous number of buffers
TARGET_SIZE	NUMBER	If a resize is in progress (state is not STATIC), records new target size (in megabytes). If the pool is STATIC, the value in this column is the same as the current size of the pool.
TARGET_BUFFERS	NUMBER	If a resize is in progress, records new target size in terms of buffers. Otherwise, the value in this column is the same as the current number of buffers.
PREV_SIZE	NUMBER	Previous buffer pool size. If the buffer pool has never been resized, the previous size is zero.
PREV_BUFFERS	NUMBER	Previous number of buffers in the buffer pool. Value is zero if the buffer pool has never been resized.
LO_BNUM	NUMBER	Obsolete column
HI_BNUM	NUMBER	Obsolete column
LO_SETID	NUMBER	Obsolete column
HI_SETID	NUMBER	Obsolete column
SET_COUNT	NUMBER	Obsolete column

See Also: "DB_BLOCK_SIZE" on page 1-42

V$BUFFER_POOL_STATISTICS

V$BUFFER_POOL_STATISTICS displays statistics about all buffer pools available for the instance.

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier number
NAME	VARCHAR2(20)	Name of the buffer pool
SET_MSIZE	NUMBER	Buffer pool maximum set size
CNUM_REPL	NUMBER	Number of buffers on replacement list
CNUM_WRITE	NUMBER	Number of buffers on write list
CNUM_SET	NUMBER	Number of buffers in set
BUF_GOT	NUMBER	Number of buffers gotten by the set
SUM_WRITE	NUMBER	Number of buffers written by the set
SUM_SCAN	NUMBER	Number of buffers scanned in the set
FREE_BUFFER_WAIT	NUMBER	Free buffer wait statistic
WRITE_COMPLETE_WAIT	NUMBER	Write complete wait statistic
BUFFER_BUSY_WAIT	NUMBER	Buffer busy wait statistic
FREE_BUFFER_INSPECTED	NUMBER	Free buffer inspected statistic
DIRTY_BUFFERS_INSPECTED	NUMBER	Dirty buffers inspected statistic
DB_BLOCK_CHANGE	NUMBER	Database blocks changed statistic
DB_BLOCK_GETS	NUMBER	Database blocks gotten statistic
CONSISTENT_GETS	NUMBER	Consistent gets statistic
PHYSICAL_READS	NUMBER	Physical reads statistic
PHYSICAL_WRITES	NUMBER	Physical writes statistic

See Also: "DB_CACHE_SIZE" on page 1-43

V$BUFFERED_PUBLISHERS

V$BUFFERED_PUBLISHERS displays information about all buffered publishers in the instance. There is one row per queue per sender. The values are reset to zero when the database (or instance in an Oracle RAC environment) restarts.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
SENDER_NAME	VARCHAR2(30)	Name of the agent enqueuing the message (the Streams name assigned for a capture process)
SENDER_ADDRESS	VARCHAR2(1024)	Queue name and database name of the source (last propagating) queue; database name is not specified if the source queue is in the local database
SENDER_PROTOCOL	NUMBER	Protocol used by the sender's address
NUM_MSGS	NUMBER	Current number of messages that have yet to be dequeued from the buffer queue
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued into the buffered queue since the database last started (comparable to V$STREAMS_CAPTURE.TOTAL_ENQUEUED_MESSAGES)

Column	Datatype	Description
LAST_ENQUEUED_MSG	NUMBER	Most recently enqueued message identifier
UNBROWSED_MSGS	NUMBER	Number of messages that have been enqueued but not browsed
OVERSPILLED_MSGS	NUMBER	Number of messages that have been spilled but not browsed
MEMORY_USAGE	NUMBER	Percentage of the Streams pool that is being used (or 0 if there is no Streams pool)
PUBLISHER_STATE	VARCHAR2(59)	State of the publisher: ■ IN FLOW CONTROL: TOO MANY UNBROWSED MESSAGES ■ IN FLOW CONTROL: OVERSPILLED MESSAGES ■ IN FLOW CONTROL: INSUFFICIENT MEMORY AND UNBROWSED MESSAGES ■ PUBLISHING MESSAGES - Normal

V$BUFFERED_QUEUES

V$BUFFERED_QUEUES displays information about all buffered queues in the instance. There is one row per queue.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
STARTUP_TIME	DATE	Startup time
NUM_MSGS	NUMBER	Total number of messages currently in the buffered queue
SPILL_MSGS	NUMBER	Current number of overflow messages spilled to disk from the buffered queue
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued into the buffered queue since the database last started
CSPILL_MSGS	NUMBER	Cumulative total number of overflow messages spilled to disk from the buffered queue since the database last started
EXPIRED_MSGS	NUMBER	Number of expired messages

V$BUFFERED_SUBSCRIBERS

V$BUFFERED_SUBSCRIBERS displays information about the subscribers for all buffered queues in the instance. There is one row per subscriber per queue.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
SUBSCRIBER_ID	NUMBER	Internal subscriber number (for identification)
SUBSCRIBER_NAME	VARCHAR2(30)	Name of the subscriber (comparable to the Streams name for the Streams process)
SUBSCRIBER_ADDRESS	VARCHAR2(1024)	Address of the subscribing agent
PROTOCOL	NUMBER	Protocol of the subscribing agent
SUBSCRIBER_TYPE	VARCHAR2(30)	Type of the subscriber: ■ PROXY - Proxy subscriber ■ SUBSCRIBER

Column	Datatype	Description
STARTUP_TIME	DATE	Startup time
LAST_BROWSED_SEQ	NUMBER	Sequence number of the most recently browsed message for the subscriber (comparable to the number of messages in the V$STREAMS_APPLY_READER view)
LAST_BROWSED_NUM	NUMBER	Internal Message number for the most recently browsed message for the subscriber
LAST_DEQUEUED_SEQ	NUMBER	Sequence number of the most recently dequeued message for the subscriber (comparable to the number of messages in the V$STREAMS_APPLY_COORDINATOR view)
LAST_DEQUEUED_NUM	NUMBER	Internal Message number for the most recently dequeued message for the subscriber
CURRENT_ENQ_SEQ	NUMBER	Current sequence number of the most recently enqueued message for the subscriber
NUM_MSGS	NUMBER	Total number of outstanding messages currently enqueued in the buffered queue for the subscriber (includes the count of the messages overflowed to disk)
CNUM_MSGS	NUMBER	Cumulative total number of messages enqueued for the subscriber since the database last started
TOTAL_DEQUEUED_MSG	NUMBER	Total number of messages dequeued by the subscriber
TOTAL_SPILLED_MSG	NUMBER	Total number of spilled messages for the subscriber
EXPIRED_MSGS	NUMBER	Number of expired messages
MESSAGE_LAG	NUMBER	Message lag of the subscriber

V$CACHE

V$CACHE displays information from the block header of each block in the SGA of the current instance as related to particular database objects. This is a Real Application Clusters view.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V$DBFILE)
BLOCK#	NUMBER	Block number
CLASS#	NUMBER	Class number
STATUS	VARCHAR2(6)	Status of the block: ■ free - Not currently in use ■ xcur - Exclusive ■ scur - Shared current ■ cr - Consistent read ■ read - Being read from disk ■ mrec - In media recovery mode ■ irec - In instance recovery mode
XNC	NUMBER	Number of PCM x to null lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block

Column	Datatype	Description
PARTITION_NAME	VARCHAR2(30)	Name of the partition (null for non-partitioned objects)
KIND	VARCHAR2(15)	Type of the database object: ■ INDEX ■ TABLE ■ CLUSTER ■ VIEW ■ SYNONYM ■ SEQUENCE ■ PROCEDURE ■ FUNCTION ■ PACKAGE ■ NON-EXISTENT ■ PACKAGE BODY ■ TRIGGER ■ TYPE ■ TYPE BODY ■ TABLE PARTITION ■ INDEX PARTITION ■ LOB ■ LIBRARY ■ JAVA SOURCE ■ JAVA CLASS ■ JAVA RESOURCE ■ JAVA DATA ■ UNDO
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Name of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

V$CACHE_LOCK

V$CACHE_LOCK is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find filename, query DBA_DATA_FILES or V$DBFILE)
BLOCK#	NUMBER	Block number

Column	Datatype	Description
STATUS	VARCHAR2(6)	Status of the block: ■ free - Not currently in use ■ xcur - Exclusive ■ scur - Shared current ■ cr - Consistent read ■ read - Being read from disk ■ mrec - In media recovery mode ■ irec - In instance recovery mode
XNC	NUMBER	Number of parallel cache management (PCM) lock conversions due to contention with another instance
NAME	VARCHAR2(30)	Name of the database object containing the block
KIND	VARCHAR2(12)	Type of database object: ■ 1 - Index ■ 2 - Table ■ 3 - Cluster ■ 4 - View ■ 5 - Synonym ■ 6 - Sequence ■ 7 - Procedure ■ 8 - Function ■ 9 - Package ■ 10 - Nonexistent ■ 11 - Package body ■ 12 - Trigger ■ 13 - Type ■ 14 - Type body ■ 19 - Table partition ■ 20 - Index partition ■ 21 - LOB ■ 22 - Library ■ Null - Unknown
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
INDX	NUMBER	Platform-specific lock manager identifier
CLASS	NUMBER	Platform-specific lock manager identifier

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

V$CACHE_TRANSFER

`V$CACHE_TRANSFER` is identical to the `V$CACHE` view but only displays blocks that have been pinged at least once. This view contains information from the block header of each block in the SGA of the current instance as related to particular database objects. This is a Real Application Clusters view.

Column	Datatype	Description
`FILE#`	`NUMBER`	Datafile identifier number (to find the filename, query `DBA_DATA_FILES` or `V$DBFILE`)
`BLOCK#`	`NUMBER`	Block number
`CLASS#`	`NUMBER`	Class number
`STATUS`	`VARCHAR2(6)`	Status of the block: ■ `free` - Not currently in use ■ `xcur` - Exclusive ■ `scur` - Shared current ■ `cr` - Consistent read ■ `read` - Being read from disk ■ `mrec` - In media recovery mode ■ `irec` - In instance recovery mode
`XNC`	`NUMBER`	Number of PCM lock conversions due to contention with another instance. This column is obsolete and maintained for backward compatibility.
`FORCED_READS`	`NUMBER`	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
`FORCED_WRITES`	`NUMBER`	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
`NAME`	`VARCHAR2(30)`	Name of the database object containing the block
`PARTITION_NAME`	`VARCHAR2(30)`	NULL for non-partitioned objects
`KIND`	`VARCHAR2(15)`	Type of database object **See Also:** Table 7–1 on page 7-14
`OWNER#`	`NUMBER`	Owner number
`GC_ELEMENT_ADDR`	`RAW(4 | 8)`	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
`GC_ELEMENT_NAME`	`NUMBER`	The name of the lock that contains the PCM lock that is covering the buffer

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

V$CALLTAG

`V$CALLTAG` ???.

Column	Datatype	Description
`SESSION_ID`	`NUMBER`	???
`PERFORMANCE_CLASS`	`VARCHAR2(32)`	???
`WORK_REQUEST_CLASS`	`VARCHAR2(32)`	???
`HOP_COUNT`	`NUMBER`	???

Column	Datatype	Description
SERVICE_NAME	VARCHAR2(64)	???
MODULE	VARCHAR2(48)	???
ACTION	VARCHAR2(32)	???
USERNAME	VARCHAR2(30)	???
PROGRAM	VARCHAR2(48)	???

V$CIRCUIT

V$CIRCUIT contains information about virtual circuits, which are user connections to the database through dispatchers and servers.

Column	Datatype	Description
CIRCUIT	RAW(4 | 8)	Circuit address
DISPATCHER	RAW(4 | 8)	Current dispatcher process address
SERVER	RAW(4 | 8)	Current server process address
WAITER	RAW(4 | 8)	Address of the server process that is waiting for the (currently busy) circuit to become available
SADDR	RAW(4 | 8)	Address of the session bound to the circuit
STATUS	VARCHAR2(16)	Status of the circuit: ■ BREAK - currently interrupted ■ EOF - about to be removed ■ OUTBOUND - an outward link to a remote database ■ NORMAL - normal circuit into the local database
QUEUE	VARCHAR2(16)	Queue the circuit is currently on: ■ COMMON - on the common queue, waiting to be picked up by a server process ■ DISPATCHER - waiting for the dispatcher ■ SERVER - currently being serviced ■ NONE - idle circuit
MESSAGE0	NUMBER	Size in bytes of the messages in the first message buffer
MESSAGE1	NUMBER	Size in bytes of the messages in the second message buffer
MESSAGE2	NUMBER	Size in bytes of the messages in the third message buffer
MESSAGE3	NUMBER	Size in bytes of the messages in the fourth message buffer
MESSAGES	NUMBER	Total number of messages that have gone through this circuit
BYTES	NUMBER	Total number of bytes that have gone through this circuit
BREAKS	NUMBER	Total number of breaks (interruptions) for this circuit
PRESENTATION	VARCHAR2(257)	Presentation protocol used by the client and server
PCIRCUIT	RAW(4 | 8)	Address of the parent circuit

V$CLASS_CACHE_TRANSFER

V$CLASS_CACHE_TRANSFER is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
CLASS	CHAR(10)	Block class; always data block

Column	Datatype	Description
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0
CR_TRANSFER	NUMBER	Number of CR blocks transferred; always 0
CURRENT_TRANSFER	NUMBER	Number of current blocks transferred; always 0

V$CLIENT_RESULT_CACHE_STATS

V$CLIENT_RESULT_CACHE_STATS ???.

Column	Datatype	Description
CLIENT REGID	NUMBER	???
BLOCK_SIZE	NUMBER	???
BLOCK_MAX	NUMBER	???
BLOCK_COUNT	NUMBER	???
BUCKET_COUNT	NUMBER	???
CREATE_SUCC	NUMBER	???
CREATE_FAIL	NUMBER	???
FINDS	NUMBER	???
INVALIDATIONS	NUMBER	???
DELETE_INVALIDS	NUMBER	???
DELETE_VALIDS	NUMBER	???

V$CLIENT_STATS

V$CLIENT_STATS displays measures for all sessions that are active for the client identifier per instance. The statistics available in this view are a subset of those available in V$SESSTAT and V$SESS_TIME_MODEL.

Column	Datatype	Description
CLIENT_IDENTIFIER	VARCHAR2(64)	Client identifier
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2(64)	Derived statistic name from V$STATNAME and V$SESS_TIME_MODEL
VALUE	NUMBER	Cumulative value (in microseconds)

V$CLUSTER_INTERCONNECTS

V$CLUSTER_INTERCONNECTS displays one or more interconnects that are being used for cluster communication.

Column	Datatype	Description
NAME	VARCHAR2(15)	Name of the interconnect (such as eth0)
IP_ADDRESS	VARCHAR2(16)	IP address of the interconnect
IS_PUBLIC	VARCHAR2(4)	If the value is YES, the interface is known to the public. If the value is NO, the interface is known to be private. Note that if the CLUSTER_INTERCONNETS initialization parameter is also specified, then it is expected that the interconnect is private. Oracle expects cluster traffic to be run on private interconnects only. f the value is empty, it is unknown whether the interface is public or private. Oracle Corporation recommends that you set the interface for Real Application Clusters (RAC) communication in the Oracle Cluster Repository (OCR).
SOURCE	VARCHAR2(31)	Indicates where this interface was picked up from: ■ Oracle Cluster Repository - Interface was configured in the OCR and Oracle Database found the interface in the OCR ■ Operating-system dependent software - Oracle Database automatically detects this ■ CLUSTER_INTERCONNECTS parameter - This initialization parameter was set

V$CONFIGURED_INTERCONNECTS

V$CONFIGURED_INTERCONNECTS displays all the interconnects that Oracle is aware of. This view attempts to answer the question of where Oracle found the information about a specific interconnect.

Column	Datatype	Description
NAME	VARCHAR2(15)	Name of the interconnect (such as eth0)
IP_ADDRESS	VARCHAR2(16)	IP address of the interconnect
IS_PUBLIC	VARCHAR2(4)	If the value is YES, the interface is known to the public. If the value is NO, the interface is known to be private. Note that if the CLUSTER_INTERCONNETS initialization parameter is also specified, then it is expected that the interconnect is private. Oracle expects cluster traffic to be run on private interconnects only. If the value is empty, it is unknown whether the interface is public or private. Oracle Corporation recommends that you set the interface for Real Application Clusters (RAC) communication in the Oracle Cluster Repository (OCR).
SOURCE	VARCHAR2(31)	Indicates where this interface was picked up from: ■ Oracle Cluster Repository - Interface was configured in the OCR and Oracle Database found the interface in the OCR ■ Operating-system dependent software - Oracle Database automatically detects this ■ CLUSTER_INTERCONNECTS parameter - This initialization parameter was set

V$CONTEXT

V$CONTEXT displays set attributes in the current session.

Column	Datatype	Description
NAMESPACE	VARCHAR2(31)	Namespace that the attribute is in
ATTRIBUTE	VARCHAR2(31)	Name of the attribute

Column	Datatype	Description
VALUE	VARCHAR2(4000)	Value of the attribute

V$CONTROLFILE

V$CONTROLFILE displays the names of the control files.

Column	Datatype	Description
STATUS	VARCHAR2(7)	INVALID if the name cannot be determined (which should not occur); NULL if the name can be determined
NAME	VARCHAR2(513)	Name of the control file
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
BLOCK_SIZE	NUMBER	Control file block size
FILE_SIZE_BLKS	NUMBER	Control file size (in blocks)

V$CONTROLFILE_RECORD_SECTION

V$CONTROLFILE_RECORD_SECTION displays information about the control file record sections.

Column	Datatype	Description
TYPE	VARCHAR2(28)	Identifies the type of record section: ■ DATABASE ■ CKPT PROGRESS ■ REDO THREAD ■ REDO LOG ■ DATAFILE ■ FILENAME ■ TABLESPACE ■ TEMPORARY FILENAME ■ RMAN CONFIGURATION ■ LOG HISTORY ■ OFFLINE RANGE ■ ARCHIVED LOG ■ BACKUP SET ■ BACKUP PIECE ■ BACKUP DATAFILE ■ BACKUP REDOLOG ■ DATAFILE COPY ■ BACKUP CORRUPTION ■ COPY CORRUPTION ■ DELETED OBJECT ■ PROXY COPY ■ BACKUP SPFILE ■ DATABASE INCARNATION ■ FLASHBACK LOG ■ RECOVERY DESTINATION ■ INSTANCE SPACE RESERVATION ■ REMOVABLE RECOVERY FILES ■ RMAN STATUS ■ THREAD INSTANCE NAME MAPPING ■ MTTR ■ DATAFILE HISTORY
RECORD_SIZE	NUMBER	Record size in bytes
RECORDS_TOTAL	NUMBER	Number of records allocated for the section
RECORDS_USED	NUMBER	Number of records used in the section
FIRST_INDEX	NUMBER	Index (position) of the first record
LAST_INDEX	NUMBER	Index of the last record
LAST_RECID	NUMBER	Record ID of the last record

V$COPY_CORRUPTION

V$COPY_CORRUPTION displays information about datafile copy corruptions from the control file.

Column	Datatype	Description
RECID	NUMBER	Copy corruption record ID
STAMP	NUMBER	Copy corruption record stamp
COPY_RECID	NUMBER	Datafile copy record ID

Column	Datatype	Description
COPY_STAMP	NUMBER	Datafile copy record stamp
FILE#	NUMBER	Datafile number
BLOCK#	NUMBER	First block of the corrupted range
BLOCKS	NUMBER	Number of contiguous blocks in the corrupted range
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
MARKED_CORRUPT	VARCHAR2(3)	(YES | NO) If set to YES the blocks were not marked corrupted in the datafile, but were detected and marked as corrupted while making the datafile copy
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: ■ ALL ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ■ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ■ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ■ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ■ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V$CORRUPT_XID_LIST

V$CORRUPT_XID_LIST displays all corrupted XIDs.

Column	Datatype	Description
CORRUPT_XID	VARCHAR2(256)	Name of corrupt XID

V$CPOOL_CC_INFO

V$CPOOL_CC_INFO displays information about the pool-to-connection class mapping for the Database Resident Connection Pool per instance.

Column	Datatype	Description
POOL_NAME	VARCHAR2(1024)	Name of the Database Resident Connection Pool
CCLASS_NAME	VARCHAR2(1024)	Name of the connection class

V$CPOOL_CC_STATS

V$CPOOL_CC_STATS displays information about the connection class level statistics for the Database Resident Connection Pool per instance.

Column	Datatype	Description
CCLASS_NAME	VARCHAR2(1024)	Name of the connection class
NUM_REQUESTS	NUMBER	Number of session requests
NUM_HITS	NUMBER	Total number of times a session that matches with the request was found in the pool

Column	Datatype	Description
NUM_MISSES	NUMBER	Total number of times an exact match to the request was not found in the pool and a new session had to be created
NUM_WAITS	NUMBER	Total number of times session requests had to wait before getting served
WAIT_TIME	NUMBER	Reserved for future use
CLIENT_REQ_TIMEOUTS	NUMBER	Reserved for future use
NUM_AUTHENTICATIONS	NUMBER	Total number of authentications of clients done by the pool

V$CPOOL_STATS

V$CPOOL_STATS displays information about the Database Resident Connection Pool statistics for an instance.

Column	Datatype	Description
POOL_NAME	VARCHAR2(1024)	Name of the Database Resident Connection Pool
NUM_OPEN_SERVERS	NUMBER	Total number of busy and free servers in the pool (including the authentication servers)
NUM_BUSY_SERVERS	NUMBER	Total number of busy servers in the pool (not including the authentication servers)
NUM_AUTH_SERVERS	NUMBER	Number of authentication servers in the pool
NUM_REQUESTS	NUMBER	Number of client requests
NUM_HITS	NUMBER	Total number of times client requests found matching pooled servers in the pool
NUM_MISSES	NUMBER	Total number of times client requests could not find a matching pooled server in the pool
NUM_WAITS	NUMBER	Total number of client requests that had to wait due to non-availability of free pooled servers
WAIT_TIME	NUMBER	Reserved for future use
CLIENT_REQ_TIMEOUTS	NUMBER	Reserved for future use
NUM_AUTHENTICATIONS	NUMBER	Total number of authentications of clients done by the pool
NUM_PURGED	NUMBER	Total number of sessions purged by the pool
HISTORIC_MAX	NUMBER	Maximum size that the pool has ever reached

V$CR_BLOCK_SERVER

V$CR_BLOCK_SERVER displays statistics on the Global Cache Service processes (LMS) used in cache fusion.

Column	Datatype	Description
CR_REQUESTS	NUMBER	Number of CR blocks served due to remote CR block requests
CURRENT_REQUESTS	NUMBER	Number of current blocks served due to remote CR block requests CR_REQUESTS + CURRENT_REQUESTS = global cache cr clocks served (from V$SYSSTAT).
DATA_REQUESTS	NUMBER	Number of current or CR requests for data blocks
UNDO_REQUESTS	NUMBER	Number of CR requests for undo blocks
TX_REQUESTS	NUMBER	Number of CR requests for undo segment header blocks DATA_REQUESTS + UNDO_REQUESTS + TX_REQUESTS = total number of requests handled by the LMS processes
OTHER_REQUESTS	NUMBER	???

Column	Datatype	Description
CURRENT_RESULTS	NUMBER	Number of requests for which no changes were rolled out of the block returned to the requesting instance
PRIVATE_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance, and only the requesting transaction can use the resulting CR block
ZERO_RESULTS	NUMBER	Number of requests for which changes were rolled out of the block returned to the requesting instance. Only zero-XID transactions can use the block.
DISK_READ_RESULTS	NUMBER	Number of requests for which the requesting instance had to read the requested block from disk
FAIL_RESULTS	NUMBER	Number of requests that failed; the requesting transaction must reissue the request
STALE	NUMBER	???
FAIRNESS_DOWN_ CONVERTS	NUMBER	Number of times an instance receiving a request has down-converted an X lock on a block because it was not modifying the block
FAIRNESS_CLEARS	NUMBER	Number of times the "fairness counter" was cleared. This counter tracks the number of times a block was modified after it was served.
FREE_GC_ELEMENTS	NUMBER	Number of times a request was received from another instance and the X lock had no buffers
FLUSHES	NUMBER	Number of times the log has been flushed by an LMS process
FLUSHES_QUEUED	NUMBER	Number of flushes queued by an LMS process
FLUSH_QUEUE_FULL	NUMBER	Number of times the flush queue was full
FLUSH_MAX_TIME	NUMBER	Maximum time for flush
LIGHT_WORKS	NUMBER	Number of times the light-work rule was evoked. This rule prevents the LMS processes from going to disk while responding to CR requests for data, undo, or undo segment header blocks. This rule can prevent the LMS process from completing its response to the CR request.
ERRORS	NUMBER	Number of times an error was signalled by an LMS process

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system for more information on these processes and on cache fusion

Note: This view contains internal diagnostic information for use by Oracle Support Services. It is subject to change without notice.

V$CURRENT_BLOCK_SERVER

V$CURRENT_BLOCK_SERVER displays statistics on the Global Cache Service processes (lMS) used in cache fusion.

Column	Datatype	Description
PIN1	NUMBER	Pins taking less than 1 millisecond
PIN10	NUMBER	Pins taking 1 to 10 milliseconds
PIN100	NUMBER	Pins taking 10 to 100 milliseconds
PIN1000	NUMBER	Pins taking 100 to 1000 milliseconds
PIN10000	NUMBER	Pins taking 1000 to 10000 milliseconds
FLUSH1	NUMBER	Flushes taking less than 1 millisecond
FLUSH10	NUMBER	Flushes taking 1 to 10 milliseconds

Column	Datatype	Description
FLUSH100	NUMBER	Flushes taking 10 to 100 milliseconds
FLUSH1000	NUMBER	Flushes taking 100 to 1000 milliseconds
FLUSH10000	NUMBER	Flushes taking 1000 to 10000 milliseconds
WRITE1	NUMBER	Writes taking less than 1 millisecond
WRITE10	NUMBER	Writes taking 1 to 10 milliseconds
WRITE100	NUMBER	Writes taking 10 to 100 milliseconds
WRITE1000	NUMBER	Writes taking 100 to 1000 milliseconds
WRITE10000	NUMBER	Writes taking 1000 to 10000 milliseconds

V$DATABASE

V$DATABASE displays information about the database from the control file.

Column	Datatype	Description
DBID	NUMBER	Database identifier calculated when the database is created and stored in all file headers
NAME	VARCHAR2(9)	Name of the database
CREATED	DATE	Creation date of the database
RESETLOGS_CHANGE#	NUMBER	System change number (SCN) at open resetlogs
RESETLOGS_TIME	DATE	Timestamp of open resetlogs
PRIOR_RESETLOGS_CHANGE#	NUMBER	SCN at prior resetlogs
PRIOR_RESETLOGS_TIME	DATE	Timestamp of prior resetlogs
LOG_MODE	VARCHAR2(12)	Archive log mode: ▪ NOARCHIVELOG ▪ ARCHIVELOG ▪ MANUAL
CHECKPOINT_CHANGE#	NUMBER	Last SCN checkpointed
ARCHIVE_CHANGE#	NUMBER	Database force archiving SCN. Any redo log with a start SCN below this will be forced to archive out.
CONTROLFILE_TYPE	VARCHAR2(7)	Type of control file: ▪ STANDBY - Indicates that the database is in standby mode ▪ CLONE - Indicates a clone database ▪ BACKUP | CREATED - Indicates the database is being recovered using a backup or created control file ▪ CURRENT - database is available for general use
CONTROLFILE_CREATED	DATE	Creation date of the control file
CONTROLFILE_SEQUENCE#	NUMBER	Control file sequence number incremented by control file transactions
CONTROLFILE_CHANGE#	NUMBER	Last SCN in backup control file; null if the control file is not a backup
CONTROLFILE_TIME	DATE	Last timestamp in backup control file; null if the control file is not a backup
OPEN_RESETLOGS	VARCHAR2(11)	(NOT ALLOWED | ALLOWED | REQUIRED) Indicates whether the next database open allows or requires the resetlogs option
VERSION_TIME	DATE	Version time

Column	Datatype	Description
OPEN_MODE	VARCHAR2(10)	Open mode information: ■ MOUNTED ■ READ WRITE ■ READ ONLY
PROTECTION_MODE	VARCHAR2(20)	Protection mode currently in effect for the database: ■ MAXIMUM PROTECTION - Database is running in maximized protection mode ■ MAXIMUM AVAILABILITY - Database is running in maximized availability mode ■ RESYNCHRONIZATION - Database is running in resynchronization mode ■ MAXIMUM PERFORMANCE - Database is running in maximized protection mode ■ UNPROTECTED - Database is unprotected (this normally occurs when the primary database is mounted and not open)
PROTECTION_LEVEL	VARCHAR2(20)	Aggregated protection mode currently in effect for the database: ■ MAXIMUM PROTECTION - Database is running in maximized protection mode ■ MAXIMUM AVAILABILITY - Database is running in maximized availability mode ■ RESYNCHRONIZATION - Database is running in resynchronization mode ■ MAXIMUM PERFORMANCE - Database is running in maximized protection mode ■ UNPROTECTED - Database is unprotected (this normally occurs when the primary database is mounted and not open) **Note:** This column is an aggregation of the PROTECTION_MODE of all standby archive log destinations.
REMOTE_ARCHIVE	VARCHAR2(8)	Value of the REMOTE_ARCHIVE_ENABLE initialization parameter
ACTIVATION#	NUMBER	Number assigned to the database instantiation
SWITCHOVER#	NUMBER	Number assigned to the database switchover
DATABASE_ROLE	VARCHAR2(16)	Current role of the database: ■ SNAPSHOT STANDBY ■ LOGICAL STANDBY ■ PHYSICAL STANDBY ■ PRIMARY
ARCHIVELOG_CHANGE#	NUMBER	Highest NEXT_CHANGE# (from the V$ARCHIVED_LOG view) for an archive log
ARCHIVELOG_COMPRESSION	VARCHAR2(8)	Status of the archive log compression (ENABLED) or (DISABLED)

Column	Datatype	Description
SWITCHOVER_STATUS	VARCHAR2(20)	Indicates whether switchover is allowed: ■ NOT ALLOWED - Either this is a standby database and the primary database has not been switched first or this is a primary database and there are no standby databases. ■ SESSIONS ACTIVE - Indicates that there are active SQL sessions attached to the primary or standby database that need to be disconnected before the switchover operation is permitted. Query the V$SESSION view to identify the specific processes that need to be terminated. ■ SWITCHOVER PENDING - This is a standby database and the primary database switchover request has been received but not processed. ■ SWITCHOVER LATENT - The switchover was in pending mode, but did not complete and went back to the primary database. ■ TO PRIMARY - This is a standby database and is allowed to switch over to a primary database. ■ TO STANDBY - This is a primary database and is allowed to switch over to a standby database. ■ RECOVERY NEEDED - This is a standby database that has not received the switchover request. ■ PREPARING SWITCHOVER - Either this is a primary database that is accepting redo data from a logical standby database in preparation for switch over to the logical standby database role, or it is a logical standby database sending redo data to a primary database and other standby databases in preparation for switch over to the primary database role. In the latter case, a completed dictionary has already been sent to the primary database and other standby databases. ■ PREPARING DICTIONARY - This is a logical standby database that is sending redo data to a primary database and other standby databases in the configuration in preparation for switch over to the primary database role. ■ TO LOGICAL STANDBY - This is a primary database that has received a complete dictionary from a logical standby database.
DATAGUARD_BROKER	VARCHAR2(8)	Indicates whether the Data Guard configuration is being managed by the broker (ENABLED) or not (DISABLED)
GUARD_STATUS	VARCHAR2(7)	Protects data from being changed: ■ ALL - Indicates all users other than SYS are prevented from making changes to any data in the database. ■ STANDBY - Indicates all users other than SYS are prevented from making changes to any database object being maintained by logical standby. ■ NONE - Indicates normal security for all data in the database.
SUPPLEMENTAL_LOG_DATA_MIN	VARCHAR2(8)	Ensures that LogMiner (and any products building on LogMiner technology) will have sufficient information to support chained rows and various storage arrangements such as cluster tables: ■ NO - None of the database-wide supplemental logging directives are enabled ■ IMPLICIT - Minimal supplemental logging is enabled because all or a combination of primary key, unique key, and foreign key supplemental logging is enabled ■ YES - Minimal supplemental logging is enabled through an ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement **See Also:** *Oracle Database SQL Language Reference* for additional information about the ALTER DATABASE ADD SUPPLEMENTAL LOG DATA statement
SUPPLEMENTAL_LOG_DATA_PK	VARCHAR2(3)	For all tables with a primary key, indicates whether all columns of the primary key are placed into the redo log whenever an update is performed (YES) or not (NO) **See Also:** *Oracle Database SQL Language Reference* for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG *supplemental_id_key_clause* statement

Column	Datatype	Description
SUPPLEMENTAL_LOG_DATA_UI	VARCHAR2(3)	For all tables with a unique key, indicates whether all other columns belonging to the unique key are placed into the redo log if any of the unique key columns are modified (YES) or not (NO) **See Also:** *Oracle Database SQL Language Reference* for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG *supplemental_id_key_clause* statement
FORCE_LOGGING	VARCHAR2(3)	Indicates whether the database is under force logging mode (YES) or not (NO)
PLATFORM_ID	NUMBER	Platform identification number of the database
PLATFORM_NAME	VARCHAR2(101)	Platform name of the database
RECOVERY_TARGET_INCARNATION#	NUMBER	Incarnation number where all datafiles are recovered by the RECOVER DATABASE command
LAST_OPEN_INCARNATION#	NUMBER	Record number of the incarnation in V$DATABASE_INCARNATION that was last opened successfully
CURRENT_SCN	NUMBER	Current SCN; null if the database is not currently open. For a standby database, the current standby redo application SCN.
FLASHBACK_ON	VARCHAR2(18)	Possible values are as follows: ■ YES - Flashback is on ■ NO - Flashback is off ■ RESTORE POINT ONLY - Flashback is on but one can only flashback to guaranteed restore points
SUPPLEMENTAL_LOG_DATA_FK	VARCHAR2(3)	For all tables with a foreign key, indicates whether all other columns belonging to the foreign key are placed into the redo log if any foreign key columns are modified (YES) or not (NO) **See Also:** *Oracle Database SQL Language Reference* for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG *supplemental_id_key_clause* statement
SUPPLEMENTAL_LOG_DATA_ALL	VARCHAR2(3)	For all columns, indicates whether all the fixed-length maximum size columns of that row are placed into the redo log (YES) or not (NO) **See Also:** *Oracle Database SQL Language Reference* for more information about the ALTER DATABASE ADD SUPPLEMENTAL LOG *supplemental_id_key_clause* statement
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name
STANDBY_BECAME_PRIMARY_SCN	NUMBER	SCN at which a physical standby database became a primary database. This SCN is useful for converting a failed primary database into a physical standby database after a forced failover. **See Also:** *Oracle Data Guard Concepts and Administration*
FS_FAILOVER_STATUS	VARCHAR2(21)	Fast-start failover status: ■ DISABLED ■ BYSTANDER ■ SYNCHRONIZED ■ UNSYNCHRONIZED ■ SUSPENDED ■ STALLED ■ LOADING DICTIONARY ■ PRIMARY UNOBSERVED ■ REINSTATE REQUIRED ■ REINSTATE IN PROGRESS ■ REINSTATE FAILED ■ TARGET OVER LAG LIMIT ■ TARGET UNDER LAG LIMIT **See Also:** *Oracle Data Guard Broker* for detailed descriptions of these values

Column	Datatype	Description
FS_FAILOVER_CURRENT_TARGET	VARCHAR2(30)	DB_UNIQUE_NAME of the standby that is the current FSFO target standby for the Data Guard configuration
FS_FAILOVER_THRESHOLD	NUMBER	Time (in seconds) that the observer will attempt to reconnect with a disconnected primary before attempting FSFO with the target standby
FS_FAILOVER_OBSERVER_PRESENT	VARCHAR2(7)	Indicates whether the observer is currently connected to the local database (YES) or not (NO) **Note:** This column is consistent throughout an Oracle RAC environment; that is, if the observer is connected to any instance, then all instances will show a value of YES.
FS_FAILOVER_OBSERVER_HOST	VARCHAR2(512)	Machine name that is currently hosting the observer process
CONTROLFILE_CONVERTED	VARCHAR2(3)	??? (YES) or (NO)
PRIMARY_DB_UNIQUE_NAME	VARCHAR2(30)	For any Standby database (Physical, Logical, or Reporting), this column will contain the DB_UNIQUE_NAME of the Primary database that this Standby last received current redo from. If this standby has not received any current redo since last being started, then this column will be null. For a Primary database that had previously been a Standby, this column will contain the DB_UNIQUE_NAME of the last Primary that this database received current redo from while acting as a Standby. For a Primary database that has never been a Standby, this column will be null.
SUPPLEMENTAL_LOG_DATA_PL	VARCHAR2(3)	??? (YES) or (NO)
MIN_REQUIRED_CAPTURE_CHANGE#	VARCHAR2(40)	Minimum REQUIRED_CHECKPOINT_SCN for all local capture processes on the database

V$DATABASE_BLOCK_CORRUPTION

V$DATABASE_BLOCK_CORRUPTION displays information about database blocks that were corrupted after the last backup.

Column	Datatype	Description
FILE#	NUMBER	Absolute file number of the datafile that contains the corrupt blocks
BLOCK#	NUMBER	Block number of the first corrupt block in the range of corrupted blocks
BLOCKS	NUMBER	Number of corrupted blocks found starting with BLOCK#
CORRUPTION_CHANGE#	NUMBER	Change number at which the logical corruption was detected. Set to 0 to indicate media corruption.
CORRUPTION_TYPE	VARCHAR2(9)	Type of block corruption in the datafile: ■ ALL ZERO - Block header on disk contained only zeros. The block may be valid if it was never filled and if it is in an Oracle7 file. The buffer will be reformatted to the Oracle8 standard for an empty block. ■ FRACTURED - Block header looks reasonable, but the front and back of the block are different versions. ■ CHECKSUM - optional check value shows that the block is not self-consistent. It is impossible to determine exactly why the check value fails, but it probably fails because sectors in the middle of the block are from different versions. ■ CORRUPT - Block is wrongly identified or is not a data block (for example, the data block address is missing) ■ LOGICAL - Specifies the range is for logically corrupt blocks. CORRUPTION_CHANGE# will have a nonzero value.

V$DATABASE_INCARNATION

V$DATABASE_INCARNATION displays information about all database incarnations. Oracle creates a new incarnation whenever a database is opened with the RESETLOGS option. Records about the current and immediately previous incarnation are also contained in the V$DATABASE view.

Column	Datatype	Description
INCARNATION#	NUMBER	Record ID for the branch record in the control file
RESETLOGS_CHANGE#	NUMBER	Resetlogs system change number (SCN) for the incarnation of the current row
RESETLOGS_TIME	DATE	Resetlogs timestamp for the incarnation of the current row
PRIOR_RESETLOGS_CHANGE#	NUMBER	Resetlogs SCN for the previous incarnation
PRIOR_RESETLOGS_TIME	DATE	Resetlogs timestamp for the previous incarnation
STATUS	VARCHAR2(7)	Incarnation status: ▪ ORPHAN - Orphan incarnation ▪ CURRENT - Current incarnation of the database ▪ PARENT - Parent of the current incarnation
RESETLOGS_ID	NUMBER	Branch ID for the incarnation of the current row (used by user-managed recovery/RMAN restore to get unique names for archived logs across incarnations)
PRIOR_INCARNATION#	NUMBER	Parent incarnation record ID if nonzero
FLASHBACK_DATABASE_ALLOWED	VARCHAR2(26)	Indicate whether or not Flashback Database can be performed into SCNs or timestamps in the incarnation. A value of YES means that you can flashback to some point in that incarnation. A value of NO indicates that you cannot flashback into the incarnation.

V$DATAFILE

V$DATAFILE displays datafile information from the control file.

See Also: "V$DATAFILE_HEADER" on page 6-67, which displays information from datafile headers

Column	Datatype	Description
FILE#	NUMBER	File identification number
CREATION_CHANGE#	NUMBER	Change number at which the datafile was created
CREATION_TIME	DATE	Timestamp of the datafile creation
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
STATUS	VARCHAR2(7)	Type of file (system or user) and its status. Values: OFFLINE, ONLINE, SYSTEM, RECOVER, SYSOFF (an offline file from the SYSTEM tablespace)
ENABLED	VARCHAR2(10)	Describes how accessible the file is from SQL: ▪ DISABLED - No SQL access allowed ▪ READ ONLY - No SQL updates allowed ▪ READ WRITE - Full access allowed ▪ UNKNOWN - should not occur unless the control file is corrupted
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Timestamp of the checkpoint#

Column	Datatype	Description
UNRECOVERABLE_CHANGE#	NUMBER	Last unrecoverable change number made to this datafile. If the database is in ARCHIVELOG mode, then this column is updated when an unrecoverable operation completes. If the database is not in ARCHIVELOG mode, this column does not get updated.
UNRECOVERABLE_TIME	DATE	Timestamp of the last unrecoverable change. This column is updated only if the database is in ARCHIVELOG mode.
LAST_CHANGE#	NUMBER	Last change number made to this datafile (null if the datafile is being changed)
LAST_TIME	DATE	Timestamp of the last change
OFFLINE_CHANGE#	NUMBER	Offline change number of the last offline range. This column is updated only when the datafile is brought online.
ONLINE_CHANGE#	NUMBER	Online change number of the last offline range
ONLINE_TIME	DATE	Online timestamp of the last offline range
BYTES	NUMBER	Current datafile size (in bytes); 0 if inaccessible
BLOCKS	NUMBER	Current datafile size (in blocks); 0 if inaccessible
CREATE_BYTES	NUMBER	Size when created (in bytes)
BLOCK_SIZE	NUMBER	Block size of the datafile
NAME	VARCHAR2(513)	Name of the datafile
PLUGGED_IN	NUMBER	Describes whether the tablespace is plugged in. The value is 1 if the tablespace is plugged in and has not been made read/write, 0 if not.
BLOCK1_OFFSET	NUMBER	Offset from the beginning of the file to where the Oracle generic information begins. The exact length of the file can be computed as follows: BYTES + BLOCK1_OFFSET.
AUX_NAME	VARCHAR2(513)	Auxiliary name that has been set for this file via CONFIGURE AUXNAME
FIRST_NONLOGGED_SCN	NUMBER	First nonlogged SCN
FIRST_NONLOGGED_TIME	DATE	First nonlogged time
FOREIGN_DBID	NUMBER	???
FOREIGN_CREATION_CHANGE#	NUMBER	???
FOREIGN_CREATION_TIME	DATE	???
PLUGGED_READONLY	VARCHAR2(3)	???
PLUGIN_CHANGE#	NUMBER	???
PLUGIN_RESETLOGS_CHANGE#	NUMBER	???
PLUGIN_RESETLOGS_TIME	DATE	???

V$DATAFILE_COPY

V$DATAFILE_COPY displays datafile copy information from the control file.

Column	Datatype	Description
RECID	NUMBER	Datafile copy record ID
STAMP	NUMBER	Datafile copy record stamp
NAME	VARCHAR2(513)	Filename of the datafile copy. The maximum length of the name is OS dependent.
TAG	VARCHAR2(32)	Datafile copy tag
FILE#	NUMBER	Absolute datafile number
RFILE#	NUMBER	Tablespace relative datafile number

Column	Datatype	Description
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made
INCREMENTAL_LEVEL	NUMBER	The incremental level
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change seen when the datafile was copied
RECOVERY_FUZZY_CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY_TIME	DATE	Timestamp of the highest change written to the file by media recovery
ONLINE_FUZZY	VARCHAR2(3)	(YES | NO) If set to YES, this is a copy taken using an operating system utility after a crash or offline immediate (or an invalid copy taken while datafile was online and the database open). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent.
BACKUP_FUZZY	VARCHAR2(3)	(YES | NO) If set to YES, this is a copy taken using the BEGIN BACKUP/END BACKUP technique. Recovery will need to apply all redo up to the end backup marker to make this copy consistent.
MARKED_CORRUPT	NUMBER	Number of blocks marked corrupt by this copy operation. That is, blocks that were not marked corrupted in the source datafile, but were detected and marked as corrupted during the copy operation.
MEDIA_CORRUPT	NUMBER	Total number of media corrupt blocks. For example, blocks with checksum errors are marked media corrupt.
LOGICALLY_CORRUPT	NUMBER	Total number of logically corrupt blocks. For example, applying redo for unrecoverable operations will mark affected blocks logically corrupt.
BLOCKS	NUMBER	Size of the datafile copy in blocks (also the size of the datafile when the copy was made)
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE_RANGE	NUMBER	RECID of the oldest offline range record in this control file copy; 0 for datafile copies
DELETED	VARCHAR2(3)	(YES | NO) If set to YES the datafile copy has been deleted or overwritten
STATUS	VARCHAR2(1)	Identifies the status of this datafile copy. Possible values are: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the copy was completed
CONTROLFILE_TYPE	VARCHAR2(1)	B indicates normal copies S indicates standby copies
KEEP	VARCHAR2(3)	(YES/NO) Indicates whether or not this backup set has a retention policy that is different than the value for the configure retention policy
KEEP_UNTIL	DATE	If KEEP_UNTIL_TIME is specified, this is the date after which the backup becomes obsolete. If this column is null, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(10)	Lists additional retention options for this backup set. Possible values are: LOGS - The logs need to recover this backup are kept NOLOGS - The logs needed to recover this backup will not be kept
SCANNED	VARCHAR2(3)	Indicates whether RMAN scanned the file (YES) or not (NO)
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)

Column	Datatype	Description
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
CONVERTED_FILE	VARCHAR2(3)	???
SAME_ENDIAN	VARCHAR2(3)	???
FOREIGN_DB	NUMBER	???
PLUGGED_READONLY	VARCHAR2(3)	???
PLUGIN_CHANGE#	NUMBER	???
PLUGIN_RESETLOGS_CHANGE#	NUMBER	???
PLUGIN_RESETLOGS_TIME	DATE	???
BACKED_BY_VSS	VARCHAR2(3)	Whether or not the file has been backed up by Volume Shadow Copy Service (VSS). This column is for internal use only.

V$DATAFILE_HEADER

V$DATAFILE_HEADER displays datafile information from the datafile headers.

Column	Datatype	Description
FILE#	NUMBER	Datafile number (from control file)
STATUS	VARCHAR2(7)	ONLINE | OFFLINE (from control file)
ERROR	VARCHAR2(18)	NULL if the datafile header read and validation were successful. If the read failed then the rest of the columns are NULL. If the validation failed then the rest of columns may display invalid data. If there is an error then usually the datafile must be restored from a backup before it can be recovered or used.
FORMAT	NUMBER	Indicates the format for the header block. The possible values are 6, 7, 8, or 0. 6 - indicates Oracle Version 6 7 - indicates Oracle Version 7 8 - indicates Oracle Version 8 0 - indicates the format could not be determined (for example, the header could not be read)
RECOVER	VARCHAR2(3)	File needs media recovery (YES | NO)
FUZZY	VARCHAR2(3)	File is fuzzy (YES | NO)
CREATION_CHANGE#	NUMBER	Datafile creation change#
CREATION_TIME	DATE	Datafile creation timestamp
TABLESPACE_NAME	VARCHAR2(30)	Tablespace name
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Tablespace relative datafile number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change#
RESETLOGS_TIME	DATE	Resetlogs timestamp
CHECKPOINT_CHANGE#	NUMBER	Datafile checkpoint change#
CHECKPOINT_TIME	DATE	Datafile checkpoint timestamp
CHECKPOINT_COUNT	NUMBER	Datafile checkpoint count
BYTES	NUMBER	Current datafile size in bytes
BLOCKS	NUMBER	Current datafile size in blocks
NAME	VARCHAR2(513)	Datafile name

Column	Datatype	Description
SPACE_HEADER	VARCHAR2(40)	The amount of space currently being used and the amount that is free, as identified in the space header
LAST_DEALLOC_SCN	VARCHAR2(16)	Last deallocated SCN

V$DATAGUARD_CONFIG

V$DATAGUARD_CONFIG displays the unique database names defined with the DB_UNIQUE_NAME and LOG_ARCHIVE_CONFIG initialization parameters, providing a view of the Data Guard environment from any database in the configuration.

The first row of the view lists the unique database name of the current database that was specified with the DB_UNIQUE_NAME initialization parameter. Additional rows reflect the unique database names of the other databases in the configuration that were specified with the DG_CONFIG keyword of the LOG_ARCHIVE_CONFIG initialization parameter.

Column	Datatype	Description
DB_UNIQUE_NAME	VARCHAR2(30)	Unique database name

V$DATAGUARD_STATS

V$DATAGUARD_STATS displays how much redo data generated by the primary database is not yet available on the standby database, showing how much redo data could be lost if the primary database were to crash at the time you queried this view. You can query this view on any instance of a standby database in a Data Guard configuration. If you query this view on a primary database, then the column values are cleared.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the parameter: ■ APPLY FINISH TIME - Estimated time before log apply services will finish applying the redo data available on the standby database. If there are gaps in the log files, then this parameter shows the time it will take to resolve the gap with the lowest SCN. ■ APPLY LAG - Amount of time that the application of redo data on the standby database lags behind the primary database. ■ TRANSPORT LAG - How much redo data (in time units) generated by the primary database is not available or applicable on the standby database at the time of computation. **Note:** This parameter includes the redo data that has not been transmitted to the standby database and redo data that is available on the standby database but has not been applied yet, such as the redo data for resolving a gap. ■ ESTIMATED STARTUP TIME - Estimated time the physical standby database was started. This is not displayed for logical standby databases. ■ STANDBY HAS BEEN OPEN - Indicates if the physical standby database was opened in read-only mode at any time since the last time the database was started. The parameter value indicates if the instance must be stopped and restarted after a failover. If the physical standby database has been opened in read-only mode since the last time it was started and it is the target of a failover, then you must shut down the database and restart it in read/write mode. If the primary database must be shut down and restarted, then use the sum of APPLY FINISH TIME and ESTIMATED STARTUP TIME to obtain an estimated failover time (the estimate assumes SHUTDOWN ABORT).

Column	Datatype	Description
VALUE	VARCHAR2(64)	Value of the parameter. For example, the APPLY FINISH TIME parameter value +00 00:00:01.7 indicates the standby database needs 1.7 seconds to finish applying the remaining redo data.
UNIT	VARCHAR2(30)	Time unit for the parameter, displayed as day (2) to second (*n*) interval, where *n* is 0 or 1.
TIME_COMPUTED	VARCHAR2(30)	Local time on the standby database when the value of the parameter was computed or sampled. The APPLY LAG and TRANSPORT LAG metrics are computed based on information received from the primary database. An unchanging value in this column for the APPLY LAG and TRANSPORT LAG metrics indicates that these metrics are not being updated (that is, they have become stale), possibly due to a communications fault between the primary and standby.

V$DATAGUARD_STATUS

V$DATAGUARD_STATUS displays messages recently written to the alert log or server process trace files that concern physical standby databases or redo transport services for all standby database types.

Column	Datatype	Description
FACILITY	VARCHAR2(24)	Facility that encountered the event: ▪ Crash Recovery ▪ Log Transport Services ▪ Log Apply Services ▪ Role Management Services ▪ Remote File Server ▪ Fetch Archive Log ▪ Data Guard ▪ Network Services
SEVERITY	VARCHAR2(13)	Severity of the event: ▪ Informational - Informational message ▪ Warning - Warning message ▪ Error - Indicates the process has failed ▪ Fatal ▪ Control - An expected change in state such as the start or completion of an archival, log recovery, or switchover operation
DEST_ID	NUMBER	Destination ID number to which the event pertains. If the event does not pertain to a particular destination, then the value is 0.
MESSAGE_NUM	NUMBER	A chronologically increasing number giving each event a unique number
ERROR_CODE	NUMBER	Error ID pertaining to the event
CALLOUT	VARCHAR2(3)	Reserved for future use
TIMESTAMP	DATE	Message date
MESSAGE	VARCHAR2(256)	A text message describing the event

V$DB_CACHE_ADVICE

V$DB_CACHE_ADVICE contains rows that predict the number of physical reads for the cache size corresponding to each row. The rows also compute a "physical read factor," which is the ratio of the number of estimated reads to the number of reads actually performed by the real buffer cache during the measurement interval.

See Also: "DB_CACHE_ADVICE" on page 1-42

Column	Datatype	Description
ID	NUMBER	Buffer pool identifier (ranges from 1 to 8)
NAME	VARCHAR2(20)	Buffer pool name
BLOCK_SIZE	NUMBER	Block size in bytes for buffers in this pool. Possible values: the standard block size, the power of 2 non-standard block sizes, 2048, 4096, 8192, 16384, 32768.
ADVICE_STATUS	VARCHAR2(3)	Status of the advisory. ON indicates it is currently running; OFF indicates it is disabled (in this case the estimates are historical and calculated when last enabled).
SIZE_FOR_ESTIMATE	NUMBER	Cache size for prediction (in megabytes)
SIZE_FACTOR	NUMBER	Size factor with respect to the current cache size
BUFFERS_FOR_ESTIMATE	NUMBER	Cache size for prediction (in terms of buffers)
ESTD_PHYSICAL_READ_FACTOR	NUMBER	Physical read factor for this cache size, which is the ratio of the number of estimated physical reads to the number of reads in the real cache. If there are no physical reads in the real cache, the value of this column is null.
ESTD_PHYSICAL_READS	NUMBER	Estimated number of physical reads for this cache size
ESTD_PHYSICAL_READ_TIME	NUMBER	Estimated disk read time
ESTD_PCT_OF_DB_TIME_FOR_READS	NUMBER	Estimated disk time as a percentage of the total time
ESTD_CLUSTER_READS	NUMBER	Estimated total number of blocks foreground processes read from the global cache (Real Application Clusters only)
ESTD_CLUSTER_READ_TIME	NUMBER	Estimated total amount of time, in seconds, foreground processes read from global cache (Real Application Clusters only)

V$DB_OBJECT_CACHE

V$DB_OBJECT_CACHE displays database objects that are cached in the library cache. Objects include tables, indexes, clusters, synonym definitions, PL/SQL procedures and packages, and triggers.

Column	Datatype	Description
OWNER	VARCHAR2(64)	Owner of the object
NAME	VARCHAR2(1000)	Name of the object
DB_LINK	VARCHAR2(64)	Database link name, if any
NAMESPACE	VARCHAR2(28)	Library cache namespace of the object: TABLE/PROCEDURE, BODY, TRIGGER, INDEX, CLUSTER, OBJECT
TYPE	VARCHAR2(28)	Type of the object: INDEX, TABLE, CLUSTER, VIEW, SET, SYNONYM, SEQUENCE, PROCEDURE, FUNCTION, PACKAGE, PACKAGE BODY, TRIGGER, CLASS, OBJECT, USER, DBLINK
SHARABLE_MEM	NUMBER	Amount of sharable memory in the shared pool consumed by the object
LOADS	NUMBER	Number of times the object has been loaded. This count also increases when an object has been invalidated.
EXECUTIONS	NUMBER	Not used **See Also:** "V$SQLAREA" on page 8-46 to see actual execution counts
LOCKS	NUMBER	Number of users currently locking this object
PINS	NUMBER	Number of users currently pinning this object
KEPT	VARCHAR2(3)	(YES | NO) Depends on whether this object has been "kept" (permanently pinned in memory) with the PL/SQL procedure DBMS_SHARED_POOL.KEEP
CHILD_LATCH	NUMBER	Child latch number that is protecting the object

Column	Datatype	Description
INVALIDATIONS	NUMBER	Total number of times objects in the namespace were marked invalid because a dependent object was modified

V$DB_PIPES

V$DB_PIPES displays the pipes that are currently represented in the shared pool for this instance.

Column	Datatype	Description
OWNERID	NUMBER	Owner ID of the owner (if this is a private pipe), else NULL
NAME	VARCHAR2(1000)	Name of the pipe (for example, SCOTT.PIPE)
TYPE	VARCHAR2(7)	Type of the pipe: ▪ PUBLIC ▪ PRIVATE
PIPE_SIZE	NUMBER	Amount of memory the pipe uses

V$DB_TRANSPORTABLE_PLATFORM

V$DB_TRANSPORTABLE_PLATFORM displays all platforms to which the database can be transported using the RMAN CONVERT DATABASE command. The transportable database feature only supports transports of the same endian platform. Therefore, V$DB_TRANSPORTABLE_PLATFORM displays fewer rows than V$TRANSPORTABLE_PLATFORM.

Column	Datatype	Description
PLATFORM_ID	NUMBER	Platform identification number
PLATFORM_NAME	VARCHAR2(101)	Platform name
ENDIAN_FORMAT	VARCHAR2(14)	Platform endian format: ▪ Big ▪ Little ▪ UNKNOWN FORMAT

V$DBFILE

V$DBFILE displays all datafiles making up the database. This view is retained for historical compatibility. Use of V$DATAFILE is recommended instead.

Column	Datatype	Description
FILE#	NUMBER	File identifier
NAME	VARCHAR2(513)	Name of the file

See Also: "V$DATAFILE" on page 6-64

V$DBLINK

V$DBLINK describes all database links (links with IN_TRANSACTION = YES) opened by the session issuing the query on V$DBLINK. These database links must be committed or rolled back before being closed.

Column	Datatype	Description
DB_LINK	VARCHAR2(128)	Name of the database link
OWNER_ID	NUMBER	Owner of the database link UID
LOGGED_ON	VARCHAR2(3)	Whether the database link is currently logged on
HETEROGENEOUS	VARCHAR2(3)	Whether the database link is heterogeneous
PROTOCOL	VARCHAR2(6)	Communication protocol for the database link
OPEN_CURSORS	NUMBER	Whether there are open cursors for the database link
IN_TRANSACTION	VARCHAR2(3)	Whether the database link is currently in a transaction
UPDATE_SENT	VARCHAR2(3)	Whether there has been an update on the database link
COMMIT_POINT_STRENGTH	NUMBER	Commit point strength of the transactions on the database link

V$DELETED_OBJECT

V$DELETED_OBJECT displays information about deleted archived logs, datafile copies and backup pieces from the control file. The only purpose of this view is to optimize the recovery catalog resync operation. When an archived log, datafile copy, or backup piece is deleted, the corresponding record is marked deleted.

Column	Datatype	Description
RECID	NUMBER	Deleted object record ID
STAMP	NUMBER	Deleted object record stamp
TYPE	VARCHAR2(26)	Identifies the type of deleted object: ▪ ARCHIVED LOG ▪ BACKUP PIECE ▪ DATAFILE COPY ▪ PROXY COPY ▪ BACKUP PIECE AVAILABLE ▪ BACKUP PIECE EXPIRED ▪ PROXY COPY AVAILABLE ▪ PROXY COPY EXPIRED ▪ BACKUP PIECE UNAVAILABLE ▪ PROXY COPY UNAVAILABLE ▪ DATAFILE COPY AVAILABLE ▪ DATAFILE COPY EXPIRED ▪ DATAFILE COPY UNAVAILABLE ▪ ARCHIVED LOG AVAILABLE ▪ ARCHIVED LOG EXPIRED ▪ ARCHIVED LOG UNAVAILABLE ▪ BACKUP SET KEEP OPTIONS ▪ BACKUP SET KEEP UNTIL ▪ PROXY COPY KEEP OPTIONS ▪ PROXY COPY KEEP UNTIL ▪ DATAFILE COPY KEEP OPTIONS ▪ DATAFILE COPY KEEP UNTIL ▪ DATAFILE RENAME ON RESTORE
OBJECT_RECID	NUMBER	Record ID of the deleted object
OBJECT_STAMP	NUMBER	Record timestamp of the deleted object
OBJECT_DATA	NUMBER	Displays additional internal information related to this deleted object. For internal Oracle use only.

Column	Datatype	Description
SET_STAMP	NUMBER	Set stamp of the deleted object
SET_COUNT	NUMBER	Set count of the deleted object

V$DISPATCHER

V$DISPATCHER displays information about the dispatcher processes.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the dispatcher process
NETWORK	VARCHAR2(128)	Network address of the dispatcher
PADDR	RAW(4 | 8)	Process address
STATUS	VARCHAR2(16)	Status of the dispatcher: ■ WAIT - Idle ■ SEND - Sending a message ■ RECEIVE - Receiving a message ■ CONNECT - Establishing a connection ■ DISCONNECT - Handling a disconnect request ■ BREAK - Handling a break ■ TERMINATE - In the process of terminating ■ ACCEPT - Accepting connections (no further information available) ■ REFUSE - Rejecting connections (no further information available)
ACCEPT	VARCHAR2(3)	Indicates whether the dispatcher is accepting new connections (YES) or not (NO)
MESSAGES	NUMBER	Number of messages processed by the dispatcher
BYTES	NUMBER	Size (in bytes) of messages processed by the dispatcher
BREAKS	NUMBER	Number of breaks occurring in the connection
OWNED	NUMBER	Number of circuits owned by the dispatcher
CREATED	NUMBER	Number of circuits created by the dispatcher
IDLE	NUMBER	Total idle time for the dispatcher (in hundredths of a second)
BUSY	NUMBER	Total busy time for the dispatcher (in hundredths of a second)
LISTENER	NUMBER	Most recent Oracle error number the dispatcher received from the listener
CONF_INDX	NUMBER	Zero-based index of the DISPATCHERS configuration used by the dispatcher

V$DISPATCHER_CONFIG

V$DISPATCHER_CONFIG displays information about the dispatcher configurations and their attributes.

Column	Datatype	Description
CONF_INDX	NUMBER	Zero-based index of the DISPATCHERS configuration
NETWORK	VARCHAR2(128)	Network protocol or listening address of dispatchers (may be truncated)
DISPATCHERS	NUMBER	Number of dispatchers to maintain for the configuration
CONNECTIONS	NUMBER	Maximum number of concurrent connections per dispatcher
SESSIONS	NUMBER	Maximum number of concurrent sessions per dispatcher

Column	Datatype	Description
POOL	VARCHAR2(4)	Indicates whether Connection Pooling is on: - IN - OUT - BOTH - OFF
TICKS	NUMBER	Tick size (in seconds) for Connection Pooling timeout
INBD_TIMOUT	NUMBER	Timeout duration (in ticks) for pooled inbound connections
OUTBD_TIMOUT	NUMBER	Timeout duration (in ticks) for pooled outbound connections
MULTIPLEX	VARCHAR2(4)	Indicates whether Session Multiplexing is on: - IN - OUT - BOTH - OFF
LISTENER	VARCHAR2(1200)	Listeners to register dispatchers with (may be truncated)
SERVICE	VARCHAR2(128)	Service names supported (may be truncated)

V$DISPATCHER_RATE

V$DISPATCHER_RATE displays rate statistics for a number of activities performed by the dispatcher processes. Collected samples have an activity-specific "time-to-live" (TTL_* columns). Statistics are reported over the following two types of time intervals:

- Current statistics (CUR_ columns)

  Current statistics use samples collected over the most recent time-to-live interval.

- Historical statistics (AVG_ and most of the MAX_ columns)

  Historical statistics make use of all samples that are no longer current.

At the time of collection, a sample is current. After the time-to-live has elapsed, the sample becomes historical. Each type of activity has a specific scale (represented by the SCALE_* columns) at which the statistics are reported.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the dispatcher process
PADDR	RAW(4 | 8)	Address of the dispatcher process
CUR_LOOP_RATE	NUMBER	Rate at which the dispatcher has been iterating through its dispatching loop, reported over the past TTL_LOOPS, in iterations/SCALE_LOOPS
CUR_EVENT_RATE	NUMBER	Rate at which the dispatcher has been processing dispatcher events, reported over the past TTL_LOOPS, in events/SCALE_LOOPS. Such dispatcher events include network events and shared server requests.
CUR_EVENTS_PER_LOOP	NUMBER	Average number of events the dispatcher has been processing in each iteration through its dispatching loop, reported over the past TTL_LOOPS, in events/iteration
CUR_MSG_RATE	NUMBER	Rate at which the dispatcher has been relaying messages between clients and shared servers, reported over the past TTL_MSG, in messages/SCALE_MSG
CUR_SVR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to shared servers, reported over the past TTL_SVR_BUF, in messages/SCALE_SVR_BUF
CUR_SVR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to shared servers, reported over the past TTL_SVR_BUF, in bytes/SCALE_SVR_BUF

Column	Datatype	Description
CUR_SVR_BYTE_PER_BUF	NUMBER	Average number of data types in each buffer relayed to shared servers, reported over the past TTL_SVR_BUF, in bytes/buffer
CUR_CLT_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to clients, reported over the past TTL_CLT_BUF, in buffers/SCALE_CLT_BUF
CUR_CLT_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to clients, reported over the past TTL_CLT_BUF, in bytes/SCALE_CLT_BUF
CUR_CLT_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to clients, reported over the past TTL_CLT_BUF, in bytes/buffer
CUR_BUF_RATE	NUMBER	Rate at which the dispatcher has been relaying buffers to either clients or shared servers, reported over the past TTL_BUF, in bytes/SCALE_BUF
CUR_BYTE_RATE	NUMBER	Rate at which the dispatcher has been relaying data to either clients or shared servers, reported over the past TTL_BUF, in bytes/SCALE_BUF
CUR_BYTE_PER_BUF	NUMBER	Average number of data bytes in each buffer relayed to either clients or shared servers, reported over the past TTL_BUF, in bytes/buffer
CUR_IN_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been accepting incoming client connections, reported over the past TTL_IN_CONNECT, in connections/SCALE_IN_CONNECT
CUR_OUT_CONNECT_RATE	NUMBER	Rate at which the dispatcher has been establishing outbound connections, reported over the past TTL_OUT_CONNECT, in connections/SCALE_OUT_CONNECT
CUR_RECONNECT_RATE	NUMBER	In a connection pooling setup, the rate at which clients have been reconnecting to the dispatcher, reported over the past TTL_RECONNECT, in reconnections/SCALE_RECONNECT
MAX_LOOP_RATE	NUMBER	Maximum rate at which the dispatcher has ever iterated through its dispatching loop, reported in iterations/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENT_RATE	NUMBER	Maximum rate at which the dispatcher has ever processed dispatcher events, reported in events/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
MAX_EVENTS_PER_LOOP	NUMBER	Maximum number of events the dispatcher has ever processed in one iteration through its dispatching loop, reported in events/iteration, over the dispatcher's lifetime
MAX_MSG_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed messages between clients and shared servers, reported in messages/SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
MAX_SVR_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to shared servers, reported in buffers/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to shared servers, reported in bytes/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
MAX_SVR_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes/buffer, over the dispatcher's lifetime
MAX_CLT_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers/SCALE_CLT_BUF, over the dispatcher's life time excluding the past TTL_CLT_BUF
MAX_CLT_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to clients, reported in bytes/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the last TTL_CLT_BUF
MAX_CLT_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to a client, reported in bytes/buffer, over the dispatcher's lifetime
MAX_BUF_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed buffers to either clients or shared servers, reported in buffers/SCALE_BUF, over the dispatcher's lifetime, excluding the past TTL_BUF
MAX_BYTE_RATE	NUMBER	Maximum rate at which the dispatcher has ever relayed data to either clients or shared servers, reported in bytes/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF

Column	Datatype	Description
MAX_BYTE_PER_BUF	NUMBER	Maximum number of data bytes the dispatcher has ever relayed in one buffer to either a client or a shared server, reported in bytes/buffer, over the dispatcher's lifetime
MAX_IN_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever accepted incoming client connections, reported in connections/SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT
MAX_OUT_CONNECT_RATE	NUMBER	Maximum rate at which the dispatcher has ever established outbound connections, reported in connections/SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT
MAX_RECONNECT_RATE	NUMBER	In a connection pooling setup, the maximum rate at which clients have ever reconnected to this dispatcher, reported in reconnections/SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
AVG_LOOP_RATE	NUMBER	Historical average rate at which the dispatcher has iterated through its dispatching loop, reported in iterations/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_EVENT_RATE	NUMBER	Historical average rate at which the dispatcher has processed dispatcher events, reported in events/SCALE_LOOPS, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_EVENTS_PER_LOOP	NUMBER	Historical average number of events the dispatcher has processed in one iteration through its dispatching loop, reported in events/iteration, over the dispatcher's lifetime excluding the past TTL_LOOPS
AVG_MSG_RATE	NUMBER	Historical average rate at which the dispatcher has relayed messages between clients and shared servers, reported in messages/SCALE_MSG, over the dispatcher's lifetime excluding the past TTL_MSG
AVG_SVR_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to shared servers, reported in buffers/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to shared servers, reported in bytes/SCALE_SVR_BUF, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_SVR_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to shared servers, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_SVR_BUF
AVG_CLT_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to clients, reported in buffers/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to clients, reported in bytes/SCALE_CLT_BUF, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_CLT_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to clients, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_CLT_BUF
AVG_BUF_RATE	NUMBER	Historical average rate at which the dispatcher has relayed buffers to either clients or shared servers, reported in buffers/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_BYTE_RATE	NUMBER	Historical average rate at which the dispatcher has relayed data to either clients or shared servers, reported in bytes/SCALE_BUF, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_BYTE_PER_BUF	NUMBER	Historical average number of data bytes per buffer the dispatcher has relayed to either clients or shared servers, reported in bytes/buffer, over the dispatcher's lifetime excluding the past TTL_BUF
AVG_IN_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has accepted incoming client connections, reported in connections/SCALE_IN_CONNECT, over the dispatcher's lifetime excluding the past TTL_IN_CONNECT
AVG_OUT_CONNECT_RATE	NUMBER	Historical average rate at which the dispatcher has established outbound connections, reported in connections/SCALE_OUT_CONNECT, over the dispatcher's lifetime excluding the past TTL_OUT_CONNECT

Column	Datatype	Description
AVG_RECONNECT_RATE	NUMBER	In a connection pooling setup, the historical average rate at which clients have reconnected to this dispatcher, reported in reconnections/SCALE_RECONNECT, over the dispatcher's lifetime excluding the past TTL_RECONNECT
TTL_LOOPS	NUMBER	Time-to-live for "loops" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_MSG	NUMBER	Time-to-live for "messages" samples, reported in hundredths of a second. Default is 10 seconds.
TTL_SVR_BUF	NUMBER	Time-to-live for "buffers to servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_CLT_BUF	NUMBER	Time-to-live for "buffers to clients" samples, reported in hundredths of a second. Default is 1 second.
TTL_BUF	NUMBER	Time-to-live for "buffers to clients/servers" samples, reported in hundredths of a second. Default is 1 second.
TTL_IN_CONNECT	NUMBER	Time-to-live for "inbound connections" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_OUT_CONNECT	NUMBER	Time-to-live for "outbound connections" samples, reported in hundredths of a second. Default is 10 minutes.
TTL_RECONNECT	NUMBER	Time-to-live for "reconnections" samples, reported in hundredths of a second. Default is 10 minutes.
SCALE_LOOPS	NUMBER	Scale for "loops" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_MSG	NUMBER	Scale for "messages" statistics, reported in hundredths of a second. Default is 1 second.
SCALE_SVR_BUF	NUMBER	Scale for "buffers to servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_CLT_BUF	NUMBER	Scale for "buffers to clients" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_BUF	NUMBER	Scale for "buffers to clients/servers" statistics, reported in hundredths of a second. Default is 1/10 second.
SCALE_IN_CONNECT	NUMBER	Scale for "inbound connections" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_OUT_CONNECT	NUMBER	Scale for "outbound connections" statistics, reported in hundredths of a second. Default is 1 minute.
SCALE_RECONNECT	NUMBER	Scale for "reconnections" statistics, reported in hundredths of a second. Default is 1 minute.

V$DNFS_CHANNELS

V$DNFS_CHANNELS displays information about the Oracle process connections (channels) open to NFS servers.

Column	Datatype	Description
PNUM	NUMBER	Oracle process number
SVRNAME	VARCHAR2(255)	NFS server name
PATH	VARCHAR2(255)	Network path to the NFS server
CH_ID	NUMBER	Direct NFS channel identifier
SVR_ID	NUMBER	Direct NFS server identifier
SENDS	NUMBER	Send operations over the channel since the last select
RECVS	NUMBER	Receive operations over the channel since the last select
PINGS	NUMBER	Ping operations over the channel since the last select

V$DNFS_FILES

V$DNFS_FILES displays information about the Oracle process files open through Direct NFS.

Column	Datatype	Description
FILENAME	VARCHAR2(513)	File name
FILESIZE	NUMBER	File Size
PNUM	NUMBER	Oracle process number which opened the file
SVR_ID	NUMBER	Direct NFS server identifier which identifies the server the file is open on

V$DNFS_SERVERS

V$DNFS_SERVERS displays information about the Direct NFS servers accessed by Direct NFS.

Column	Datatype	Description
ID	NUMBER	Direct NFS server identifier
SVRNAME	VARCHAR2(255)	NFS server name
DIRNAME	VARCHAR2(1024)	Mounted directory
MNTPORT	NUMBER	NFS mount port
NFSPORT	NUMBER	NFS port
WTMAX	NUMBER	WTMAX exported by the NFS server
RTMAX	NUMBER	RTMAX exported by the NFS server

V$DNFS_STATS

V$DNFS_STATS displays information about the Oracle process NFS operation statistics issued by Direct NFS.

Column	Datatype	Description
PNUM	NUMBER	Oracle process number that statistics are relevant to
NFS_NULL	NUMBER	Null
NFS_GETATTR	NUMBER	Get attributes
NFS_SETATTR	NUMBER	Set attributes
NFS_LOOKUP	NUMBER	Lookup object
NFS_ACCESS	NUMBER	Access object
NFS_READLINK	NUMBER	Read link
NFS_READ	NUMBER	Read file
NFS_WRITE	NUMBER	Write file
NFS_CREATE	NUMBER	Create file
NFS_MKDIR	NUMBER	Make directory
NFS_SYMLINK	NUMBER	Symbolic link
NFS_MKNOD	NUMBER	Make node
NFS_REMOVE	NUMBER	Remove file
NFS_RMDIR	NUMBER	Remove directory
NFS_RENAME	NUMBER	Rename

Column	Datatype	Description
NFS_LINK	NUMBER	Link
NFS_READDIR	NUMBER	Read directory
NFS_READDIRPLUS	NUMBER	Read directory plus
NFS_FSSTAT	NUMBER	File system status
NFS_FSINFO	NUMBER	File system information
NFS_PATHCONF	NUMBER	Path configuration
NFS_COMMIT	NUMBER	Commit
NFS_MOUNT	NUMBER	Mount

V$DYNAMIC_REMASTER_STATS

V$DYNAMIC_REMASTER_STATS displays statistical information about the dynamic remastering process. All times are given in hundredths of a second, and total values reflect what has been collected since instance startup.

Column	Datatype	Description
REMASTER_OPS	NUMBER	Total number of dynamic remastering operations
REMASTER_TIME	NUMBER	Total dynamic remastering time
REMASTERED_OBJECTS	NUMBER	Total number of objects dynamically remastered due to affinity
QUIESCE_TIME	NUMBER	Total quiesce step time
FREEZE_TIME	NUMBER	Total freeze step time
CLEANUP_TIME	NUMBER	Total cleanup step time
REPLAY_TIME	NUMBER	Total replay step time
FIXWRITE_TIME	NUMBER	Total fixwrite step time
SYNC_TIME	NUMBER	Total synchronization step time
RESOURCES_CLEANED	NUMBER	Total number of resources cleaned in the cleanup steps
REPLAYED_LOCKS_SENT	NUMBER	Total number of locks replayed to other instances in the replay steps
REPLAYED_LOCKS_RECEIVED	NUMBER	Total number of locks received from other instances in the replay steps
CURRENT_OBJECTS	NUMBER	Current number of objects remastered on this instance due to affinity

V$ENABLEDPRIVS

V$ENABLEDPRIVS displays which privileges are enabled. These privileges can be found in the table SYSTEM_PRIVILEGE_MAP.

Column	Datatype	Description
PRIV_NUMBER	NUMBER	Numeric identifier of enabled privileges

See Also: "SYSTEM_PRIVILEGE_MAP" on page 5-94

V$ENCRYPTED_TABLESPACES

V$ENCRYPTED_TABLESPACES displays information about the tablespaces that are encrypted.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
ENCRYPTIONALG	VARCHAR2(7)	Encryption algorithm: ■ NONE ■ 3DES168 ■ AES128 ■ AES192 ■ AES256
ENCRYPTEDTS	VARCHAR2(3)	Indicates whether the tablespace is encrypted (YES) or not (NO)

V$ENCRYPTION_WALLET

V$ENCRYPTION_WALLET displays information on the status of the wallet and the wallet location for transparent data encryption.

Column	Datatype	Description
WRL_TYPE	VARCHAR2(20)	Type of the wallet resource locator (for example, FILE)
WRL_PARAMETER	VARCHAR2(4000)	Parameter of the wallet resource locator (for example, absolute filename if WRL_TYPE = FILE)
STATUS	VARCHAR2(9)	Status of the wallet: ■ OPEN ■ CLOSED ■ UNDEFINED

V$ENQUEUE_LOCK

V$ENQUEUE_LOCK displays all locks owned by enqueue state objects. The columns in this view are identical to the columns in V$LOCK.

See Also: "V$LOCK" on page 7-13

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of lock state object
KADDR	RAW(4 | 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. Lists user and system types that can have locks.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)

Column	Datatype	Description
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V$ENQUEUE_STAT

V$ENQUEUE_STAT displays statistics on the number of enqueue (lock) requests for each type of lock.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
EQ_TYPE	VARCHAR2(2)	Type of enqueue requested
TOTAL_REQ#	NUMBER	Total number of enqueue requests or enqueue conversions for this type of enqueue
TOTAL_WAIT#	NUMBER	Total number of times an enqueue request or conversion resulted in a wait
SUCC_REQ#	NUMBER	Number of times an enqueue request or conversion was granted
FAILED_REQ#	NUMBER	Number of times an enqueue request or conversion failed
CUM_WAIT_TIME	NUMBER	Total amount of time (in milliseconds) spent waiting for the enqueue or enqueue conversion

V$ENQUEUE_STATISTICS

V$ENQUEUE_STATISTICS displays statistics on the number of enqueue (lock) requests for each type of lock. V$ENQUEUE_STATISTICS encompasses V$ENQUEUE_STAT and gives more detailed information (several rows for same enqueues with different reasons).

Column	Datatype	Description
EQ_NAME	VARCHAR2(64)	Name of the enqueue request
EQ_TYPE	VARCHAR2(2)	Type of enqueue requested
REQ_REASON	VARCHAR2(64)	Reason for the enqueue request
TOTAL_REQ#	NUMBER	Total number of enqueue requests or enqueue conversions for this type of enqueue
TOTAL_WAIT#	NUMBER	Total number of times an enqueue request or conversion resulted in a wait
SUCC_REQ#	NUMBER	Number of times an enqueue request or conversion was granted
FAILED_REQ#	NUMBER	Number of times an enqueue request or conversion failed
CUM_WAIT_TIME	NUMBER	Total amount of time (in milliseconds) spent waiting for the enqueue or enqueue conversion
REQ_DESCRIPTION	VARCHAR2(4000)	Description of the enqueue request
EVENT#	NUMBER	Event number

V$EVENT_HISTOGRAM

V$EVENT_HISTOGRAM displays a histogram of the number of waits, the maximum wait, and total wait time on an event basis. The histogram has buckets of time intervals from < 1 ms, < 2 ms, < 4 ms, < 8 ms, ... < 2^{21} ms, < 2^{22} ms, and >= 2^{22} ms.

The histogram will not be filled unless the TIMED_STATISTICS initialization parameter is set to true.

Column	Datatype	Description
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Name of the Event
WAIT_TIME_MILLI	NUMBER	Amount of time the bucket represents (in milliseconds). If the duration = *num*, then this column represents waits of duration < *num* that are not included in any smaller bucket.
WAIT_COUNT	NUMBER	Number of waits of the duration belonging to the bucket of the histogram

V$EVENT_NAME

V$EVENT_NAME displays information about wait events.

Column	Datatype	Description
EVENT#	NUMBER	Number of the wait event
EVENT_ID	NUMBER	Identifier of the wait event
NAME	VARCHAR2(64)	Name of the wait event
PARAMETER1	VARCHAR2(64)	Description of the first parameter for the wait event
PARAMETER2	VARCHAR2(64)	Description of the second parameter for the wait event
PARAMETER3	VARCHAR2(64)	Description of the third parameter for the wait event
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2(64)	Name of the class of the wait event **See Also:** "Classes of Wait Events" on page C-1 for a description of the different wait event classes

V$EVENTMETRIC

V$EVENTMETRIC displays values of wait event metrics for the most recent 60-second interval.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
EVENT#	NUMBER	Number of the event
EVENT_ID	NUMBER	Identifier of the event
NUM_SESS_WAITING	NUMBER	Number of sessions waiting at the end of the interval
TIME_WAITED	NUMBER	Time waited
WAIT_COUNT	NUMBER	Number of times waited

V$EXECUTION

V$EXECUTION displays information on parallel execution.

Column	Datatype	Description
PID	NUMBER	Session ID
DEPTH	NUMBER	The depth
FUNCTION	VARCHAR2(10)	Session serial number
TYPE	VARCHAR2(7)	Name of the OBJECT_NODE in plan table
NVALS	NUMBER	Elapsed time for OBJECT_NODE
VAL1	NUMBER	The value for number 1
VAL2	NUMBER	The value for number 2
SEQH	NUMBER	A sequence
SEQL	NUMBER	A sequence

V$FALSE_PING

V$FALSE_PING is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
FILE#	NUMBER	Datafile identifier number (to find the filename, query DBA_DATA_FILES or V$DBFILE)
BLOCK#	NUMBER	Block number
STATUS	VARCHAR2(6)	Status of the block: ▪ free - Not currently in use ▪ xcur - Exclusive ▪ scur - Shared current ▪ cr - Consistent read ▪ read - Being read from disk ▪ mrec - In media recovery mode ▪ irec - In instance recovery mode
XNC	NUMBER	Number of PCM lock conversions from Exclusive mode due to contention with another instance. This column is obsolete and maintained for backward compatibility.
FORCED_READS	NUMBER	Number of times the block had to be reread from the cache because another instance has forced it out of this instance's cache by requesting the lock on the block in exclusive mode
FORCED_WRITES	NUMBER	Number of times GCS had to write this block to cache because this instance had used the block and another instance had requested the lock on the block in a conflicting mode
NAME	VARCHAR2(30)	Name of the database object containing the block
PARTITION_NAME	VARCHAR2(30)	NULL for non-partitioned objects
KIND	VARCHAR2(15)	Type of database object
OWNER#	NUMBER	Owner number
LOCK_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
LOCK_ELEMENT_NAME	NUMBER	Name of the lock that contains the PCM lock that is covering the buffer
LOCK_ELEMENT_CLASS	NUMBER	Lock element class

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

V$FAST_START_SERVERS

V$FAST_START_SERVERS provides information about all the recovery slaves performing parallel transaction recovery.

Column	Datatype	Description
STATE	VARCHAR2(11)	State of the server (IDLE or RECOVERING)
UNDOBLOCKSDONE	NUMBER	Number of undo blocks done so far
PID	NUMBER	Process ID
XID	RAW(8)	Transaction ID

See Also: *Oracle Database Performance Tuning Guide*

V$FAST_START_TRANSACTIONS

V$FAST_START_TRANSACTIONS displays information about the progress of the transactions that Oracle is recovering.

Column	Datatype	Description
USN	NUMBER	Undo segment number of the transaction
SLT	NUMBER	Slot within the rollback segment
SEQ	NUMBER	Incarnation number of the slot
STATE	VARCHAR2(16)	State of the transaction (may be TO BE RECOVERED, RECOVERED, or RECOVERING)
UNDOBLOCKSDONE	NUMBER	Number of undo blocks completed on the transaction
UNDOBLOCKSTOTAL	NUMBER	Total number of undo blocks that need recovery
PID	NUMBER	ID of the current server it has been assigned to
CPUTIME	NUMBER	Time for which recovery has progressed (in seconds)
PARENTUSN	NUMBER	Undo segment number of the parent transaction in PDML
PARENTSLT	NUMBER	Slot of the parent transaction in PDML
PARENTSEQ	NUMBER	Sequence number of the parent transaction in PDML
XID	RAW(8)	Transaction ID
PXID	RAW(8)	Parent transaction ID
RCVSERVERS	NUMBER	Number of servers used in the last recovery

See Also: *Oracle Database Performance Tuning Guide*

V$FILE_CACHE_TRANSFER

V$FILE_CACHE_TRANSFER is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the datafile

Column	Datatype	Description
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
RBR	NUMBER	Number of reuse blocks cross-instance calls; always 0
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse blocks cross-instance calls; always 0
RBR_FORCED_STALE	NUMBER	Number of blocks marked as flushed due to reuse blocks cross-instance calls; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0
CR_TRANSFERS	NUMBER	Number of CR blocks received; always 0
CUR_TRANSFERS	NUMBER	Number of current blocks received; always 0

V$FILE_HISTOGRAM

V$FILE_HISTOGRAM displays a histogram of all synchronous single block reads on a per-file basis (for data files). The histogram has buckets of time intervals from < 1 ms, < 2 ms, < 4 ms, < 8 ms, ... < 2^{21} ms, < 2^{22} ms, and >= 2^{22} ms.

The histogram will not be filled unless the STATISTICS_LEVEL initialization parameter is set to ALL.

Column	Datatype	Description
FILE#	NUMBER	File number
SINGLEBLKRDTIM_MILLI	NUMBER	Amount of time the bucket represents (in milliseconds). If the duration = *num*, then this column represents waits of duration < *num* that are not included in any smaller bucket.
SINGLEBLKRDS	NUMBER	Number of waits of the duration belonging to the bucket of the histogram

V$FILEMETRIC

V$FILEMETRIC displays values of file metrics for the most recent 10-minute interval. A history of the last one hour will be kept in the system.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
FILE_ID	NUMBER	File number
CREATION_TIME	NUMBER	Timestamp of the file creation
AVERAGE_READ_TIME	NUMBER	Average file read time
AVERAGE_WRITE_TIME	NUMBER	Average file write time
PHYSICAL_READS	NUMBER	Number of physical reads

Column	Datatype	Description
PHYSICAL_WRITES	NUMBER	Number of physical writes
PHYSICAL_BLOCK_READS	NUMBER	Number of physical block reads
PHYSICAL_BLOCK_WRITES	NUMBER	Number of physical block writes

V$FILEMETRIC_HISTORY

V$FILEMETRIC_HISTORY displays values of file metrics for all intervals in the last one hour.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
FILE_ID	NUMBER	File number
CREATION_TIME	NUMBER	Timestamp of the file creation
AVERAGE_READ_TIME	NUMBER	Average file read time
AVERAGE_WRITE_TIME	NUMBER	Average file write time
PHYSICAL_READS	NUMBER	Number of physical reads
PHYSICAL_WRITES	NUMBER	Number of physical writes
PHYSICAL_BLOCK_READS	NUMBER	Number of physical block reads
PHYSICAL_BLOCK_WRITES	NUMBER	Number of physical block writes

V$FILESPACE_USAGE

V$FILESPACE_USAGE summarizes space allocation information of each datafile and tempfile.

Column	Datatype	Description
TABLESPACE_ID	NUMBER	ID of the tablespace to which the file belongs
RFNO	NUMBER	Relative file number of the file
ALLOCATED_SPACE	NUMBER	Total allocated space in the file
FILE_SIZE	NUMBER	Current file size
FILE_MAXSIZE	NUMBER	Maximum file size
CHANGESCN_BASE	NUMBER	SCN base of the last change to the file
CHANGESCN_WRAP	NUMBER	SCN wrap of the last change to the file
FLAG	NUMBER	Flags for file attributes

V$FILESTAT

V$FILESTAT displays the number of physical reads and writes done and the total number of single-block and multiblock I/Os done at file level. As of Oracle Database 10g Release 2 (10.2), this view also includes reads done by RMAN processes for backup operations.

Column	Datatype	Description
FILE#	NUMBER	Number of the file

Column	Datatype	Description
PHYRDS	NUMBER	Number of physical reads done
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
SINGLEBLKRDS	NUMBER	Number of single block reads
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER	Cumulative single block read time (in hundredths of a second)
AVGIOTIM	NUMBER	Average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is true; 0 if false
LSTIOTIM	NUMBER	Time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MINIOTIM	NUMBER	Minimum time (in hundredths of a second) spent on a single I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIORTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is true; 0 if false

V$FIXED_TABLE

V$FIXED_TABLE displays all dynamic performance tables, views, and derived tables in the database. Some V$ tables (for example, V$ROLLNAME) refer to real tables and are therefore not listed.

Column	Datatype	Description
NAME	VARCHAR2(30)	Name of the object
OBJECT_ID	NUMBER	Identifier of the fixed object
TYPE	VARCHAR2(5)	Object type (TABLE | VIEW)
TABLE_NUM	NUMBER	Number that identifies the dynamic performance table if it is of type TABLE

V$FIXED_VIEW_DEFINITION

V$FIXED_VIEW_DEFINITION contains the definitions of all the fixed views (views beginning with V$). Use this table with caution. Oracle tries to keep the behavior of fixed views the same from release to release, but the definitions of the fixed views can change without notice. Use these definitions to optimize your queries by using indexed columns of the dynamic performance tables.

Column	Datatype	Description
VIEW_NAME	VARCHAR2(30)	Name of the fixed view
VIEW_DEFINITION	VARCHAR2(4000)	Definition of the fixed view

V$FLASH_RECOVERY_AREA_USAGE

V$FLASH_RECOVERY_AREA_USAGE displays usage information about flashback recovery areas.

Column	Datatype	Description
FILE_TYPE	VARCHAR2(19)	File type: ■ CONTROL FILE ■ REDO LOG ■ ARCHIVED LOG ■ BACKUP PIECE ■ IMAGE COPY ■ FLASHBACK LOG ■ REMOTE ARCHIVED LOG
PERCENT_SPACE_USED	NUMBER	Percent of recovery area that is in use
PERCENT_SPACE_RECLAIMABLE	NUMBER	Percent of recovery area that is reclaimable
NUMBER_OF_FILES	NUMBER	Number of files in the recovery area

V$FLASHBACK_DATABASE_LOG

V$FLASHBACK_DATABASE_LOG displays information about the flashback data. Use this view to help estimate the amount of flashback space required for the current workload.

Column	Datatype	Description
OLDEST_FLASHBACK_SCN	NUMBER	Lowest system change number (SCN) in the flashback data, for any incarnation
OLDEST_FLASHBACK_TIME	DATE	Time of the lowest SCN in the flashback data, for any incarnation
RETENTION_TARGET	NUMBER	Target retention time (in minutes)
FLASHBACK_SIZE	NUMBER	Current size (in bytes) of the flashback data
ESTIMATED_FLASHBACK_SIZE	NUMBER	Estimated size of flashback data needed for the current target retention

V$FLASHBACK_DATABASE_LOGFILE

V$FLASHBACK_DATABASE_LOGFILE displays information about the flashback log files.

Column	Datatype	Description
NAME	VARCHAR2(513)	Name of the log file
LOG#	NUMBER	Log file number
THREAD#	NUMBER	Log file thread number
SEQUENCE#	NUMBER	Log file sequence number
BYTES	NUMBER	Log file size (in bytes)
FIRST_CHANGE#	NUMBER	Lowest system change number (SCN) in the log file
FIRST_TIME	DATE	Time of the first SCN in the log file

V$FLASHBACK_DATABASE_STAT

V$FLASHBACK_DATABASE_STAT displays statistics for monitoring the I/O overhead of logging flashback data. This view also displays the estimated flashback space needed based on previous workloads.

Column	Datatype	Description
BEGIN_TIME	DATE	Beginning of the time interval
END_TIME	DATE	End of the time interval
FLASHBACK_DATA	NUMBER	Number of bytes of flashback data written during the interval
DB_DATA	NUMBER	Number of bytes of database data read and written during the interval
REDO_DATA	NUMBER	Number of bytes of redo data written during the interval
ESTIMATED_FLASHBACK_SIZE	NUMBER	Value of ESTIMATED_FLASHBACK_SIZE in V$FLASHBACK_DATABASE_LOG at the end of the time interval

V$FLASHBACK_TXN_GRAPH

V$FLASHBACK_TXN_GRAPH displays a tabular representation of the transaction dependency graph. For each dependency edge, there could be multiple rows, one for each conflicting operation..

This view is relevant AFTER a compensating transaction has been started through the DBMS_FLASHBACK.TRANSACTION_BACKOUT() set of functions, and is no longer relevant once the compensating transaction is either committed or rolled back. It also provides a tabular representation of the undo SQL that is not available through the CLOB XML construct in the DBA_FLASHBACK_TXN_REPORT view.

Column	Datatype	Description
COMPENSATING_XID	RAW(8)	Transaction ID of the compensating transaction
COMPENSATING_TXN_NAME	VARCHAR2(255)	Name of the compensating transaction
XID	RAW(8)	Transaction ID of a relevant transaction found in memory
TXN_NAME	VARCHAR2(255)	Name of the transaction with XID as the transaction ID; NULL if none
PARENT_XID	RAW(8)	Parent transaction ID (for a PDML transaction)
INTERESTING	NUMBER	If the transaction is in the transaction dependency graph
ORIGINAL	NUMBER	If the transaction is part of the input set provided
BACKOUT_SEQ	NUMBER	Order in which the transaction has been backed out
NUM_PREDS	NUMBER	Number of predecessors of the transaction specified by XID in the transaction graph
NUM_SUCCS	NUMBER	Number of successors of the transaction specified by XID in the transaction graph
DEP_XID	RAW(8)	One dependent transaction ID of the transaction specified by XID. This is a particular child of XID.
DEP_TXN_NAME	VARCHAR2(255)	Transaction name, if any, for the transaction specified by DEP_XID
TXN_CONF_SQL_ID	NUMBER	SQL ID of undo SQL executed in the context of XID which conflicts with the dependent transaction
DEP_TXN_CONF_SQL_ID	NUMBER	SQL ID of undo SQL executed in the context of DEP_XID which conflicts with XID

V$FLASHBACK_TXN_MODS

V$FLASHBACK_TXN_MODS displays the individual modifications of all the transactions in memory.

This view is relevant AFTER a compensating transaction has been started through the DBMS_FLASHBACK.TRANSACTION_BACKOUT() set of functions, and is no longer relevant once the compensating transaction is either committed or rolled back. It also provides a tabular representation of the undo SQL that is not available through the CLOB XML construct in the DBA_FLASHBACK_TXN_REPORT view.

Column	Datatype	Description
COMPENSATING_XID	RAW(8)	Transaction ID of the compensating transaction
COMPENSATING_TXN_NAME	VARCHAR2(255)	Name of the compensating transaction
XID	RAW(8)	Transaction ID of a relevant transaction found in memory
TXN_NAME	VARCHAR2(255)	Name of the transaction with XID as the transaction ID; NULL if none
PARENT_XID	RAW(8)	Parent transaction ID (for a PDML transaction)
INTERESTING	NUMBER	If the transaction is in the transaction dependency graph
ORIGINAL	NUMBER	If the transaction is part of the input set provided
BACKOUT_SEQ	NUMBER	Order in which the transaction has been backed out
UNDO_SQL	VARCHAR2(4000)	Undo SQL for the modification
UNDO_SQL_SQN	NUMBER	Order in which the given SQL has been executed to back out this transaction
UNDO_SQL_SUB_SQN	NUMBER	If the undo SQL is greater than 4000 bytes, then a sequence number, starting from 1, of a 4000-byte division of the undo SQL
BACKOUT_SQL_ID	NUMBER	SQL ID of the undo SQL (used only for this compensating transaction)
OPERATION	VARCHAR2(30)	Operation (such as insert/update/delete) performed by the forward-going operation
BACKEDOUT	NUMBER	Indicates whether the transaction has been backed out as of now
CONFLICT_MOD	NUMBER	If the concerned modification is causing a conflict
MODS_PER_LCR	NUMBER	Sometimes an LCR could cause multiple modifications (for example, an update of an IOT could actually be a delete followed by an insert)

V$FOREIGN_ARCHIVED_LOG

V$FOREIGN_ARCHIVED_LOG can be queried on a logical standby database to find out the list of foreign archived logs received by a database.

Column	Datatype	Description
RECID	NUMBER	Archived log record ID
STAMP	NUMBER	Archived log record stamp
NAME	VARCHAR2(513)	Archived log file name. If set to NULL, either the log file was cleared before it was archived or an RMAN BACKUP command with the delete input option was executed to back up archivelog all (RMAN> backup archivelog all delete input;).
DEST_ID	NUMBER	Original destination from which the archive log was generated. The value is 0 if the destination identifier is not available
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when the log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when the log was written
RESETLOGS_ID	NUMBER	Resetlogs identifier associated with the archived redo log
FIRST_CHANGE#	NUMBER	First change number in the archived log
FIRST_TIME	DATE	Timestamp of the first change

Column	Datatype	Description
NEXT_CHANGE#	NUMBER	First change in the next log
NEXT_TIME	DATE	Timestamp of the next change
BLOCKS	NUMBER	Size of the archived log (in blocks)
BLOCK_SIZE	NUMBER	Redo log block size. This is the logical block size of the archived log, which is the same as the logical block size of the online log from which the archived log was copied. The online log logical block size is a platform-specific value that is not adjustable by the user.
CREATOR	VARCHAR2(7)	Creator of the archive log: ARCH - Archiver process FGRD - Foreground process RMAN - Recovery Manager SRMN - RMAN at standby LGWR - Logwriter process
REGISTRAR	VARCHAR2(7)	Registrar of the entry: RFS - Remote File Server process ARCH - Archiver process FGRD - Foreground process RMAN - Recovery Manager SRMN - RMAN at standby LGWR - Logwriter process
ARCHIVED	VARCHAR2(3)	Indicates whether the online redo log was archived (YES) or whether RMAN only inspected the log and created a record for future application of redo logs during recovery (NO). See *Oracle Database Backup and Recovery User's Guide* for more information.
APPLIED	VARCHAR2(3)	Indicates whether the archivelog has been applied to its corresponding standby database (YES) or not (NO). The value is always NO for local destinations. This column is meaningful at the physical standby site for the ARCHIVED_LOG entries with REGISTRAR='RFS' (which means this log is shipped from the primary to the standby database). If REGISTRAR='RFS' and APPLIED is NO, then the log has arrived at the standby but has not yet been applied. If REGISTRAR='RFS' and APPLIED is YES, the log has arrived and been applied at the standby database. You can use this field to identify archivelogs that can be backed up and removed from disk.
DELETED	VARCHAR2(3)	Indicates whether an RMAN DELETE command has physically deleted the archived log file from disk, as well as logically removing it from the control file of the target database and from the recovery catalog (YES) or not (NO)
STATUS	VARCHAR2(1)	Status of the archived log: A - Available D - Deleted U - Unavailable X - Expired
COMPLETION_TIME	DATE	Time when the archiving completed
DICTIONARY_BEGIN	VARCHAR2(3)	Indicates whether the log contains the start of a LogMiner dictionary (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)	Indicates whether the log contains the end of a LogMiner dictionary (YES) or not (NO)
END_OF_REDO	VARCHAR2(3)	Indicates whether the archived redo log contains the end of all redo information from the primary database () or not ()
ARCHIVAL_THREAD#	NUMBER	Redo thread number of the instance that performed the archival operation. This column differs from the THREAD# column only when a closed thread is archived by another instance.

Column	Datatype	Description
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)
COMPRESSED	VARCHAR2(3)	This column is reserved for internal use only
FAL	VARCHAR2(3)	Indicates whether the archive log was generated as the result of a FAL request (YES) or not (NO)
END_OF_REDO_TYPE	VARCHAR2(10)	Possible values are as follows: SWITCHOVER - Shows archived redo log files that are produced at the end of a switchover TERMINAL - Shows archived redo log files produced after a failover RESETLOGS - Shows online redo log files archived on the primary database after an ALTER DATABASE OPEN RESETLOGS statement is issued ACTIVATION - Shows any log files archived on a physical standby database after an ALTER DATABASE ACTIVATE STANDBY DATABASE statement is issued "empty string" - Any empty string implies that the log is just a normal archival and was not archived due to any of the other events
SOURCE_DBID	NUMBER	Database ID of the source database that generated this archived log

V$FS_FAILOVER_STATS

V$FS_FAILOVER_STATS displays statistics about Fast-Start Failovers (FSFOs) occurring on the system.

Column	Datatype	Description
LAST_FAILOVER_TIME	VARCHAR2(20)	Timestamp of the last Fast-Start Failover
LAST_FAILOVER_REASON	VARCHAR2(255)	Reason for the last Fast-Start Failover

V$GC_ELEMENT

V$GC_ELEMENT displays one entry for each global cache resource that is used by the buffer cache. The name of the global cache resource that corresponds to a lock element is {'BL', indx, class}. This is a Real Application Clusters view.

See Also: The *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your operating system.

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock that is covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.
INDX	NUMBER	Platform specific lock manager identifier
CLASS	NUMBER	Platform specific lock manager identifier
GC_ELEMENT_NAME	NUMBER	Name of the lock that contains the PCM lock that is covering the buffer
MODE_HELD	NUMBER	Platform dependent value for lock mode held; often: 3 = share; 5 = exclusive
BLOCK_COUNT	NUMBER	Number of blocks covered by PCM lock
RELEASING	NUMBER	Nonzero if PCM lock is being downgraded
ACQUIRING	NUMBER	Nonzero if PCM lock is being upgraded
INVALID	NUMBER	Nonzero if PCM lock is invalid (a lock may become invalid after a system failure)

Column	Datatype	Description
FLAGS	NUMBER	Process level flags for the lock element

V$GC_ELEMENTS_WITH_COLLISIONS

V$GC_ELEMENTS_WITH_COLLISIONS is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
GC_ELEMENT_ADDR	RAW(4 | 8)	Address of the lock element that contains the PCM lock covering the buffer. If more than one buffer has the same address, then these buffers are covered by the same PCM lock.

V$GCSHVMASTER_INFO

V$GCSHVMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Cache Service resources except those belonging to files mapped to a particular master.

Column	Datatype	Description
HV_ID	NUMBER	PCM hash value ID
CURRENT_MASTER	NUMBER	Master instance of this PCM hash value ID
PREVIOUS_MASTER	NUMBER	Previous master instance of this PCM hash value ID
REMASTER_CNT	NUMBER	Number of times this has been remastered

V$GCSPFMASTER_INFO

V$GCSPFMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Cache Service resources belonging to files mapped to instances.

Column	Datatype	Description
FILE_ID	NUMBER	File number
OBJECT_ID	NUMBER	Object ID
TYPE	NUMBER	Type of object
CURRENT_MASTER	NUMBER	Master instance of this file
PREVIOUS_MASTER	NUMBER	Previous master instance of this file
REMASTER_CNT	NUMBER	Number of times this has been remastered

V$GES_BLOCKING_ENQUEUE

V$GES_BLOCKING_ENQUEUE describes all locks currently known to lock manager that are being blocked or blocking others. The output of this view is a subset of the output from V$GES_ENQUEUE. This is a Real Application Clusters view.

See Also: "V$GES_ENQUEUE" on page 6-96 for a description of all locks known to the lock manager

Column	Datatype	Description
HANDLE	RAW(4 | 8)	Lock pointer
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier where the lock belongs to
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier where the lock belongs to
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS_OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT_GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT_PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT_NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT_DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT_NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue.
STATE	VARCHAR2(64)	State of lock as owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V$GES_CONVERT_LOCAL

V$GES_CONVERT_LOCAL displays statistics for local GES enqueue operations. This view records average convert times, count information, and timed statistics for global enqueue requests.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance

Column	Datatype	Description
CONVERT_TYPE	VARCHAR2(16)	Conversion type: ■ NULL -> SS - NULL mode to subshared mode ■ NULL -> SX - NULL mode to shared exclusive mode ■ NULL -> S - NULL mode to shared mode ■ NULL -> SSX - NULL mode to subshared exclusive mode ■ NULL -> X - NULL mode to exclusive mode ■ SS -> SX - Subshared mode to shared exclusive mode ■ SS -> S - Subshared mode to shared mode ■ SS -> SSX - Subshared mode to subshared exclusive mode ■ SS -> X - Subshared mode to exclusive mode ■ SX -> S - Shared exclusive mode to shared mode ■ SX -> SSX - Shared exclusive mode to subshared exclusive mode ■ SX -> X - Shared exclusive mode to exclusive mode ■ S -> SX - Shared mode to shared exclusive mode ■ S -> SSX - Shared mode to subshared exclusive mode ■ S -> X - Shared mode to exclusive mode ■ SSX -> X - Sub-shared exclusive mode to exclusive mode
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in hundredths of a second)
CONVERT_COUNT	NUMBER	Number of operations

V$GES_CONVERT_REMOTE

V$GES_CONVERT_REMOTE displays values for remote GES enqueue conversions. This view records average convert times, count information, and timed statistics for global enqueue requests.

Column	Datatype	Description
INST_ID	NUMBER	ID of the instance
CONVERT_TYPE	VARCHAR2(16)	Conversion type: ■ NULL -> SS - NULL mode to subshared mode ■ NULL -> SX - NULL mode to shared exclusive mode ■ NULL -> S - NULL mode to shared mode ■ NULL -> SSX - NULL mode to subshared exclusive mode ■ NULL -> X - NULL mode to exclusive mode ■ SS -> SX - Subshared mode to shared exclusive mode ■ SS -> S - Subshared mode to shared mode ■ SS -> SSX - Subshared mode to subshared exclusive mode ■ SS -> X - Subshared mode to exclusive mode ■ SX -> S - Shared exclusive mode to shared mode ■ SX -> SSX - Shared exclusive mode to subshared exclusive mode ■ SX -> X - Shared exclusive mode to exclusive mode ■ S -> SX - Shared mode to shared exclusive mode ■ S -> SSX - Shared mode to subshared exclusive mode ■ S -> X - Shared mode to exclusive mode ■ SSX -> X - Sub-shared exclusive mode to exclusive mode
AVERAGE_CONVERT_TIME	NUMBER	Average conversion time for each type of lock operation (in hundredths of a second)
CONVERT_COUNT	NUMBER	Number of operations

V$GES_ENQUEUE

V$GES_ENQUEUE describes all locks currently known to lock manager. This is a Real Application Clusters view.

See Also: "V$GES_BLOCKING_ENQUEUE" on page 6-93 for a description of all such locks that are currently blocking or being blocked

Column	Datatype	Description
HANDLE	RAW(4 | 8)	Lock pointer
GRANT_LEVEL	VARCHAR2(9)	Granted level of the lock
REQUEST_LEVEL	VARCHAR2(9)	Requested level of the lock
RESOURCE_NAME1	VARCHAR2(30)	Resource name for the lock
RESOURCE_NAME2	VARCHAR2(30)	Resource name for the lock
PID	NUMBER	Process identifier which holds the lock
TRANSACTION_ID0	NUMBER	Lower 4 bytes of the transaction identifier to which the lock belongs
TRANSACTION_ID1	NUMBER	Upper 4 bytes of the transaction identifier to which the lock belongs
GROUP_ID	NUMBER	Group identifier for the lock
OPEN_OPT_DEADLOCK	NUMBER	1 if DEADLOCK open option is set, otherwise 0
OPEN_OPT_PERSISTENT	NUMBER	1 if PERSISTENT open option is set, otherwise 0
OPEN_OPT_PROCESS_OWNED	NUMBER	1 if PROCESS_OWNED open option is set, otherwise 0
OPEN_OPT_NO_XID	NUMBER	1 if NO_XID open option is set, otherwise 0
CONVERT_OPT_GETVALUE	NUMBER	1 if GETVALUE convert option is set, otherwise 0
CONVERT_OPT_PUTVALUE	NUMBER	1 if PUTVALUE convert option is set, otherwise 0
CONVERT_OPT_NOVALUE	NUMBER	1 if NOVALUE convert option is set, otherwise 0
CONVERT_OPT_DUBVALUE	NUMBER	1 if DUBVALUE convert option is set, otherwise 0
CONVERT_OPT_NOQUEUE	NUMBER	1 if NOQUEUE convert option is set, otherwise 0
CONVERT_OPT_EXPRESS	NUMBER	1 if EXPRESS convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKWAIT	NUMBER	1 if NODEADLOCKWAIT convert option is set, otherwise 0
CONVERT_OPT_NODEADLOCKBLOCK	NUMBER	1 if NODEADLOCKBLOCK convert option is set, otherwise 0
WHICH_QUEUE	NUMBER	Which queue the lock is currently located. 0 for NULL queue; 1 for GRANTED queue; 2 for CONVERT queue.
STATE	VARCHAR2(64)	State of the lock as the owner sees it
AST_EVENT0	NUMBER	Last AST event
OWNER_NODE	NUMBER	Node identifier
BLOCKED	NUMBER	1 if this lock request is blocked by others, otherwise 0
BLOCKER	NUMBER	1 if this lock is blocking others, otherwise 0

V$GES_LATCH

V$GES_LATCH is deprecated.

See Also: "V$LATCH" on page 7-8 for statistics about GES latch performance

V$GES_RESOURCE

V$GES_RESOURCE is a Real Application Clusters view. It displays information of all resources currently known to the lock manager.

Column	Datatype	Description
RESP	RAW(4 | 8)	Resource pointer
RESOURCE_NAME	VARCHAR2(30)	Resource name in hexadecimal for the lock
ON_CONVERT_Q	NUMBER	1 if on convert queue, 0 otherwise
ON_GRANT_Q	NUMBER	1 if on granted queue, 0 otherwise
PERSISTENT_RES	NUMBER	1 if it is a persistent resource, 0 otherwise
RDOMAIN_NAME	VARCHAR2(25)	Recovery domain name
RDOMAINP	RAW(4 | 8)	Recovery domain pointer
MASTER_NODE	NUMBER	Master node ID
NEXT_CVT_LEVEL	VARCHAR2(9)	Next lock level to convert on global convert queue
VALUE_BLK_STATE	VARCHAR2(32)	State of the value block
VALUE_BLK	VARCHAR2(64)	First 64 bytes of the value block

V$GES_STATISTICS

V$GES_STATISTICS displays miscellaneous GES statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Name of the statistic
VALUE	NUMBER	Value associated with the statistic

V$GLOBAL_BLOCKED_LOCKS

V$GLOBAL_BLOCKED_LOCKS displays global blocked locks.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of lock state object
KADDR	RAW(4 | 8)	Address of lock
SID	NUMBER	Identifier of session holding the lock (number)
TYPE	VARCHAR2(2)	Resource type (char)
ID1	NUMBER	Resource identifier #1 (number)
ID2	NUMBER	Resource identifier #2 (number)
LMODE	NUMBER	Lock mode held (number)
REQUEST	NUMBER	Lock mode requested (number)
CTIME	NUMBER	Time since current mode was granted

V$GLOBAL_TRANSACTION

V$GLOBAL_TRANSACTION displays information on the currently active global transactions.

Column	Datatype	Description
FORMATID	NUMBER	Format identifier of the global transaction
GLOBALID	RAW(64)	Global transaction identifier of the global transaction
BRANCHID	RAW(64)	Branch qualifier of the global transaction
BRANCHES	NUMBER	Total number of branches in the global transaction
REFCOUNT	NUMBER	Number of siblings for the global transaction (must be the same as branches)
PREPARECOUNT	NUMBER	Number of branches of the global transaction that have prepared
STATE	VARCHAR2(38)	State of the branch of the global transaction: ▪ ACTIVE ▪ COLLECTING ▪ FINALIZED ▪ FAILED ▪ RECOVERING ▪ UNASSOCIATED ▪ FORGOTTEN ▪ READY FOR RECOVERY ▪ NO-READONLY FAILED ▪ SIBLING INFO WRITTEN ▪ [ORACLE COORDINATED]ACTIVE ▪ [ORACLE COORDINATED]COLLECTING ▪ [ORACLE COORDINATED]FINALIZED ▪ [ORACLE COORDINATED]FAILED ▪ [ORACLE COORDINATED]RECOVERING ▪ [ORACLE COORDINATED]UNASSOCIATED ▪ [ORACLE COORDINATED]FORGOTTEN ▪ [ORACLE COORDINATED]READY FOR RECOVERY ▪ [ORACLE COORDINATED]NO-READONLY FAILED ▪ [MULTINODE]ACTIVE ▪ [MULTINODE]COLLECTING ▪ [MULTINODE]FINALIZED ▪ [MULTINODE]FAILED ▪ [MULTINODE]RECOVERING ▪ [MULTINODE]UNASSOCIATED ▪ [MULTINODE]FORGOTTEN ▪ [MULTINODE]READY FOR RECOVERY ▪ [MULTINODE]NO-READONLY FAILED ▪ [MULTINODE]SIBLING INFO WRITTEN ▪ COMBINATION
FLAGS	NUMBER	The numerical representation of the state
COUPLING	VARCHAR2(15)	Indicates whether the branches are free (FREE), loosely coupled (LOOSELY COUPLED), or tightly coupled (TIGHTLY COUPLED)

V$HM_CHECK

V$HM_CHECK displays information about all the checks registered with Health Monitor. Each check is uniquely identified by a name or an ID.

Column	Datatype	Description
ID	NUMBER	Health check ID
NAME	VARCHAR2(64)	Health check name
CLSID	NUMBER	Class ID to which this check belongs
CLS_NAME	VARCHAR2(15)	Class name of the check: ■ GENERIC ■ PERSISTENT_DATA
FLAGS	NUMBER	Reserved for internal use
INTERNAL_CHECK	VARCHAR2(1)	Internal check
OFFLINE_CAPABLE	VARCHAR2(1)	Ability to run when database is not open yet (Y or N)
DESCRIPTION	VARCHAR2(64)	Description of what the check does

V$HM_CHECK_PARAM

V$HM_CHECK_PARAM displays information about the input parameters of all Health Monitor checks.

Column	Datatype	Description
ID	NUMBER	Parameter ID
NAME	VARCHAR2(64)	Parameter name
CHECK_ID	NUMBER	ID of the check to which this parameter belongs
TYPE	VARCHAR2(20)	Data type of the input parameter: ■ DBKH_PARAM_UB4 ■ DBKH_PARAM_UB8 ■ DBKH_PARAM_TEXT ■ DBKH_PARAM_DATE ■ DBKH_PARAM_UB4_LIST ■ DBKH_PARAM_UB8_LIST ■ DBKH_PARAM_TEXT_LIST ■ DBKH_PARAM_DATE_LIST
DEFAULT_VALUE	VARCHAR2(64)	Default value for this parameter
FLAGS	NUMBER	Reserved for internal use
DESCRIPTION	VARCHAR2(64)	Description of the parameter

V$HM_FINDING

V$HM_FINDING displays information about all the findings of various Health Monitor runs.

Column	Datatype	Description
FINDING_ID	NUMBER	Unique ID to represent the finding
RUN_ID	NUMBER	ID of the run that created this finding
NAME	VARCHAR2(32)	Name of the finding
PARENT_ID	NUMBER	Parent finding ID for this finding
CHILD_COUNT	NUMBER	Number of active (open) child findings, if this finding is a parent finding type
CLASS_NAME	VARCHAR2(32)	Name of the class to which this finding belongs

Column	Datatype	Description
TIME_DETECTED	TIMESTAMP(6)	Time this finding was detected
MODIFIED	TIMESTAMP(6)	Time that this finding was last modified
PRIORITY	VARCHAR2(8)	Priority of the finding: ■ LOW, ■ HIGH ■ CRITICAL
STATUS	VARCHAR2(12)	Status of the finding: ■ OPEN ■ CLOSED
TYPE	VARCHAR2(13)	Type of the finding: ■ INFORMATIONAL ■ FAILURE
DESCRIPTION	VARCHAR2(1024)	Description of the finding
DAMAGE_DESCRIPTION	VARCHAR2(1024)	Possible damage description of the finding

V$HM_INFO

V$HM_INFO displays information about Health Monitor runs, findings, and recommendations. The information for a run/finding/recommendation is organized as a name, value pair.

If the type of information is RUN, then the data represents the input parameters for that run. If the type of information is FINDING or RECOMMENDATION, then the data represents the information about that particular finding/recommendation.

Column	Datatype	Description
ID	NUMBER	Unique identifier of the information
TYPE	VARCHAR2(14)	Type of the information: ■ RUN ■ RUN-RESUME ■ FINDING ■ RECOMMENDATION
NAME	VARCHAR2(32)	Information parameter name
VALUE	VARCHAR2(513)	Information parameter value

V$HM_RECOMMENDATION

V$HM_RECOMMENDATION displays information about all the recommendations made to various Health Monitor findings.

Column	Datatype	Description
RECOMMENDATION_ID	NUMBER	Unique ID to represent the recommendation
FDG_ID	NUMBER	Unique ID to represent either the finding ID or the finding set ID for which the recommendation was made
RUN_ID	NUMBER	ID of the run that may have generated the recommendation If the ID is 0, then the recommendations were generated by RMAN.
NAME	VARCHAR2(32)	Name of the recommendation

Column	Datatype	Description
TYPE	VARCHAR2(7)	Type of the recommendation: ■ MANUAL ■ REPAIR
RANK	NUMBER	Rank of the recommendation
TIME_DETECTED	TIMESTAMP(6)	Time that the recommendation was made
EXECUTED	TIMESTAMP(6)	Time that the recommendation (repair) was implemented
STATUS	VARCHAR2(7)	Status of the recommendation (repair) execution: ■ NOT RUN ■ RUNNING ■ SUCCESS ■ FAILED
DESCRIPTION	VARCHAR2(1024)	Description of the recommendation
REPAIR_SCRIPT	VARCHAR2(512)	Location of the repair script file

V$HM_RUN

V$HM_RUN displays information about all Health Monitor checks and their status.

Column	Datatype	Description
RUN_ID	NUMBER	Unique ID to represent the run
NAME	VARCHAR2(32)	Unique name used to identify the run
CHECK_NAME	VARCHAR2(32)	Name of the check
RUN_MODE	VARCHAR2(8)	Mode of the run: ■ MANUAL ■ REACTIVE ■ AUTO
TIMEOUT	NUMBER	Number of seconds allowed for this run to complete before it is aborted
START_TIME	TIMESTAMP(6)	Start time of the run
LAST_RESUME_TIME	TIMESTAMP(6)	Last resumed time for the run
END_TIME	TIMESTAMP(6)	End time of the run
MODIFIED_TIME	TIMESTAMP(6)	Last modified time of the run record
STATUS	VARCHAR2(11)	Status of the run: ■ INITIAL ■ EXECUTING ■ INTERRUPTED ■ TIMEDOUT ■ CANCELLED ■ COMPLETED ■ ERROR
SRC_INCIDENT	NUMBER	Source incident ID that activated this run
NUM_INCIDENT	NUMBER	Number of incidents created by this run
ERROR_NUMBER	NUMBER	Error number if the run failed to complete because of an error
PROBLEM_ID	NUMBER	Problem ID of the source incident ID that may have activated this Run

V$HS_AGENT

V$HS_AGENT displays the set of HS agents currently running on a given host. There is one row per agent process.

Column	Datatype	Description
AGENT_ID	NUMBER	Oracle Net session identifier used for connections to the agent (the identifier used in the LISTENER.ORA file)
MACHINE	VARCHAR2(64)	Operating system machine name
PROCESS	VARCHAR2(9)	Operating system process identifier of the agent
PROGRAM	VARCHAR2(48)	Program name of the agent
OSUSER	VARCHAR2(30)	Operating system user
STARTTIME	DATE	Starting time
AGENT_TYPE	NUMBER	Type of the agent
FDS_CLASS_ID	NUMBER	ID of the Foreign Data Store class
FDS_INST_ID	NUMBER	Instance name of the Foreign Data Store

V$HS_PARAMETER

V$HS_PARAMETER describes the initialization parameters in use by the server and agent.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier (maps to the HS_SESSION_ID column of V$HS_SESSION)
PARAMETER	VARCHAR2(30)	Name of the parameter
VALUE	VARCHAR2(64)	Value of the parameter
SOURCE	VARCHAR2(1)	Indicates whether the parameter was defined in the agent (A) or server (S)
ENV	VARCHAR2(1)	Indicates whether the parameter was set as an environment variable in the agent (T) or elsewhere (F)

V$HS_SESSION

V$HS_SESSION describes the current HS session.

Column	Datatype	Description
HS_SESSION_ID	NUMBER	Unique HS session identifier
AGENT_ID	NUMBER	Oracle Net session identifier used for connections to the agent (maps to the AGENT_ID column of V$HS_AGENT)
SID	NUMBER	User session identifier (maps to the SID column of V$SESSION)
DB_LINK	VARCHAR2(128)	Server database link name used to access the agent; blank if no database link is used (for example, when using external procedures)
DB_LINK_OWNER	NUMBER	Owner of the database link in DB_LINK
STARTTIME	DATE	Time the connection was initiated

V$HVMASTER_INFO

V$HVMASTER_INFO describes the current and previous master instances and the number of re-masterings of Global Enqueue Service resources.

Column	Datatype	Description
HV_ID	NUMBER	Hash value ID
CURRENT_MASTER	NUMBER	Master instance of this hash value ID
PREVIOUS_MASTER	NUMBER	Previous master instance of this hash value ID
REMASTER_CNT	NUMBER	Number of times this has been remastered

7

Dynamic Performance (V$) Views: V$INCMETER_CONFIG to V$RULE_SET_AGGREGATE_STATS

This chapter contains the dynamic performance views V$INCMETER_CONFIG to V$RULE_SET_AGGREGATE_STATS.

V$INDEXED_FIXED_COLUMN

V$INDEXED_FIXED_COLUMN displays the columns in dynamic performance tables that are indexed (X$ tables). The X$ tables can change without notice. Use this view only to write queries against fixed views (V$ views) more efficiently.

Column	Datatype	Description
TABLE_NAME	VARCHAR2(30)	Name of the dynamic performance table that is indexed
INDEX_NUMBER	NUMBER	Number that distinguishes to which index a column belongs
COLUMN_NAME	VARCHAR2(30)	Name of the column that is being indexed
COLUMN_POSITION	NUMBER	Position of the column in the index key (this is mostly relevant for multicolumn indexes)

V$INSTANCE

V$INSTANCE displays the state of the current instance.

Column	Datatype	Description
INSTANCE_NUMBER	NUMBER	Instance number used for instance registration (corresponds to the INSTANCE_NUMBER initialization parameter) **See Also:** "INSTANCE_NUMBER" on page 1-71
INSTANCE_NAME	VARCHAR2(16)	Name of the instance
HOST_NAME	VARCHAR2(64)	Name of the host machine
VERSION	VARCHAR2(17)	Database version
STARTUP_TIME	DATE	Time when the instance was started
STATUS	VARCHAR2(12)	Status of the instance: ▪ STARTED - After STARTUP NOMOUNT ▪ MOUNTED - After STARTUP MOUNT or ALTER DATABASE CLOSE ▪ OPEN - After STARTUP or ALTER DATABASE OPEN ▪ OPEN MIGRATE - After ALTER DATABASE OPEN { UPGRADE | DOWNGRADE }

Column	Datatype	Description
PARALLEL	VARCHAR2(3)	Indicates whether the instance is mounted in cluster database mode (YES) or not (NO)
THREAD#	NUMBER	Redo thread opened by the instance
ARCHIVER	VARCHAR2(7)	Automatic archiving status: ■ STOPPED ■ STARTED ■ FAILED - Archiver failed to archive a log last time but will try again within 5 minutes
LOG_SWITCH_WAIT	VARCHAR2(15)	Event that log switching is waiting for: ■ ARCHIVE LOG ■ CLEAR LOG ■ CHECKPOINT ■ NULL - ALTER SYSTEM SWITCH LOGFILE is hung but there is room in the current online redo log
LOGINS	VARCHAR2(10)	Indicates whether the instance is in unrestricted mode, allowing logins by all users (ALLOWED, or in restricted mode, allowing logins by database administrators only (RESTRICTED)
SHUTDOWN_PENDING	VARCHAR2(3)	Indicates whether a shutdown is pending (YES) or not (NO)
DATABASE_STATUS	VARCHAR2(17)	Status of the database: ■ ACTIVE ■ SUSPENDED ■ INSTANCE RECOVERY
INSTANCE_ROLE	VARCHAR2(18)	Indicates whether the instance is an active instance (PRIMARY_INSTANCE) or an inactive secondary instance (SECONDARY_INSTANCE), or UNKNOWN if the instance has been started but not mounted
ACTIVE_STATE	VARCHAR2(9)	Quiesce state of the instance: ■ NORMAL - Database is in a normal state. ■ QUIESCING - ALTER SYSTEM QUIESCE RESTRICTED has been issued: no new user transactions, queries, or PL/SQL statements are processed in this instance. User transactions, queries, or PL/SQL statements issued before the ALTER SYSTEM QUIESCE RESTRICTED statement are unaffected. DBA transactions, queries, or PL/SQL statements are also unaffected. ■ QUIESCED - ALTER SYSTEM QUIESCE RESTRICTED has been issued: no user transactions, queries, or PL/SQL statements are processed. DBA transactions, queries, or PL/SQL statements are unaffected. User transactions, queries, or PL/ SQL statements issued after the ALTER SYSTEM QUIESCE RESTRICTED statement are not processed. A single ALTER SYSTEM QUIESCE RESTRICTED statement quiesces all instances in an Oracle RAC environment. After this statement has been issued, some instances may enter into a quiesced state before other instances; the system is quiesced when all instances enter the quiesced state.
BLOCKED	VARCHAR2(3)	Indicates whether all services are blocked (YES) or not (NO)

V$INSTANCE_CACHE_TRANSFER

V$INSTANCE_CACHE_TRANSFER displays statistics for the cache blocks transferred among instances.

Oracle keeps multiple versions of data buffered in the buffer cache. The current buffer (or block), CURRENT_BLOCK, is the most up-to-date copy, containing all recent modifications. A consistent read buffer (or block), CR_BLOCK, contains the version of the data at a particular time prior to the current buffer. It is read-consistent (that is, all the data shown in that buffer are consistent for the start time of a query).

Therefore, for the same data block there can be multiple copies in the buffer cache: one current copy, and one or more consistent read copies with data consistent as of different snapshot times.

Column	Datatype	Description
INSTANCE	NUMBER	Instance from which the blocks are transferred
CLASS	VARCHAR2(18)	Class of the cache block
LOST	NUMBER	The number of blocks that were sent by a particular instance but that never arrived in this instance
LOST_TIME	NUMBER	The time waited for blocks that were sent by a particular instance but that never arrived in this instance
CR_BLOCK	NUMBER	CR Block transfers not affected by remote processing delays
CR_BLOCK_TIME	NUMBER	Total time waited for cr blocks from a particular instance (includes the other times)
CR_2HOP	NUMBER	The count of cr blocks which were received by this instance from a particular instance after a 2-way round-trip
CR_2HOP_TIME	NUMBER	The time waited for cr blocks which were received by this instance from a particular instance after a 2-way round-trip
CR_3HOP	NUMBER	The count of cr blocks which were received by this instance from a particular instance after a 3-way round-trip
CR_3HOP_TIME	NUMBER	The time waited for cr blocks which were received by this instance from a particular instance after a 3-way round-trip
CR_BUSY	NUMBER	CR Block transfers affected by remote contention
CR_BUSY_TIME	NUMBER	The time waited for cr blocks which were received by this instance from a particular instance and which were delayed by a log flushed on the sending instance
CR_CONGESTED	NUMBER	CR Block transfers affected by remote system load
CR_CONGESTED_TIME	NUMBER	The time waited for cr blocks which were received by this instance from a particular instance and which were delayed becaused LMS was busy
CURRENT_BLOCK	NUMBER	Current block transfers not affected by remote processing delays
CURRENT_BLOCK_TIME	NUMBER	Total time waited for current blocks from a particular instance (includes the other times
CURRENT_2HOP	NUMBER	The count of current blocks which were received by this instance from a particular instance after a 2-way roundtrip
CURRENT_2HOP_TIME	NUMBER	The time waited for current blocks which were received by this instance from a particular instance after a 2-way roundtrip
CURRENT_3HOP	NUMBER	The count of current blocks which were received by this instance from a particular instance after a 3-way roundtri
CURRENT_3HOP_TIME	NUMBER	The time waited for current blocks which were received by this instance from a particular instance after a 3-way roundtrip
CURRENT_BUSY	NUMBER	Current block transfers affected by remote contention
CURRENT_BUSY_TIME	NUMBER	The time waited for current blocks which were received by this instance from a particular instance and which were delayed by a log flushed on the sending instance
CURRENT_CONGESTED	NUMBER	Current block transfers affected by remote system load
CURRENT_CONGESTED_TIME	NUMBER	The time waited for current blocks which were received by this instance from a particular instance and which were delayed becaused LMS was busy

V$INSTANCE_RECOVERY

V$INSTANCE_RECOVERY monitors the mechanisms available to users to limit recovery I/O. Those mechanisms are:

- Set the LOG_CHECKPOINT_TIMEOUT initialization parameter
- Set the LOG_CHECKPOINT_INTERVAL initialization parameter
- Set the FAST_START_MTTR_TARGET initialization parameter
- Set the size of the smallest redo log

Column	Datatype	Description
RECOVERY_ESTIMATED_IOS	NUMBER	Number of dirty buffers in the buffer cache. In the Standard Edition, this column is always null.
ACTUAL_REDO_BLKS	NUMBER	Current actual number of redo blocks required for recovery
TARGET_REDO_BLKS	NUMBER	Current target number of redo blocks that must be processed for recovery. This value is the minimum value of the following 3 columns, and identifies which of the 3 user-defined limits determines checkpointing.
LOG_FILE_SIZE_REDO_BLKS	NUMBER	Maximum number of redo blocks required to guarantee that a log switch does not occur before the checkpoint completes.
LOG_CHKPT_TIMEOUT_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_TIMEOUT parameter. The value displayed is not meaningful unless that parameter has been set.
LOG_CHKPT_INTERVAL_REDO_BLKS	NUMBER	Number of redo blocks that need to be processed during recovery to satisfy the LOG_CHECKPOINT_INTERVAL parameter. The value displayed is not meaningful unless that parameter has been set.
FAST_START_IO_TARGET_REDO_BLKS	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always null.
TARGET_MTTR	NUMBER	Effective MTTR (mean time to recover) target value in seconds. The TARGET_MTTR value is calculated based on the value of the FAST_START_MTTR_TARGET parameter (the TARGET_MTTR value is used internally), and is usually an approximation of the parameter's value. However, if the FAST_START_MTTR_TARGET parameter value is very small (for example, one second), or very large (for example, 3600 seconds), the calculation will produce a target value dictated by system limitations. In such cases, the TARGET_MTTR value will be the shortest calculated time, or the longest calculated time that recovery is expected to take. If FAST_START_MTTR_TARGET is not specified, the value of this field is the current estimated MTTR.
ESTIMATED_MTTR	NUMBER	Current estimated mean time to recover (MTTR) based on the number of dirty buffers and log blocks (0 if FAST_START_MTTR_TARGET is not specified). Basically, this value tells you how long you could expect recovery to take based on the work your system is doing right now.
CKPT_BLOCK_WRITES	NUMBER	Number of blocks written by checkpoint writes
OPTIMAL_LOGFILE_SIZE	NUMBER	Redo log file size (in megabytes) that is considered optimal based on the current setting of FAST_START_MTTR_TARGET. It is recommended that the user configure all online redo logs to be at least this value. Note that redo log files must be at least 4 megabytes in size; otherwise an error is generated.
ESTD_CLUSTER_AVAILABLE_TIME	NUMBER	Estimated time (in seconds) that the cluster would become partially available should this instance fail. This column is only meaningful in a Real Application Clusters (RAC) environment. In a non-RAC environment, the value of this column is null.
WRITES_MTTR	NUMBER	Number of writes driven by the FAST_START_MTTR_TARGET initialization parameter
WRITES_LOGFILE_SIZE	NUMBER	Number of writes driven by the smallest redo log file size
WRITES_LOG_CHECKPOINT_SETTINGS	NUMBER	Number of writes driven by the LOG_CHECKPOINT_INTERVAL or LOG_CHECKPOINT_TIMEOUT initialization parameter
WRITES_OTHER_SETTINGS	NUMBER	Number of writes driven by other reasons (such as the deprecated FAST_START_IO_TARGET initialization parameter)
WRITES_AUTOTUNE	NUMBER	Number of writes due to auto-tune checkpointing

Column	Datatype	Description
WRITES_FULL_THREAD_ CKPT	NUMBER	Number of writes due to full thread checkpoints

See Also:

- *Oracle Database Performance Tuning Guide* for more information on limiting recovery I/O, and how Oracle determines MTTR recovery times
- LOG_CHECKPOINT_INTERVAL on page 1-91
- FAST_START_MTTR_TARGET on page 1-62

V$IO_CALIBRATION_STATUS

V$IO_CALIBRATION_STATUS displays the status of I/O calibration in the instance.

Column	Datatype	Description
STATUS	VARCHAR2(13)	Calibration status: ■ IN PROGRESS - Calibration in Progress (Results from a previous calibration run are displayed, if available) ■ READY Results are ready and available from an earlier run ■ NOT AVAILABLE Calibration results are not available
CALIBRATION_TIME	TIMESTAMP(3)	End time of the last calibration run

V$IOFUNCMETRIC

V$IOFUNCMETRIC displays I/O statistics information by database function for the most recent time interval period.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time for the metric value
END_TIME	DATE	End time for the metric value
INTSIZE_CSEC	NUMBER	Size of the time period
FUNCTION_ID	NUMBER	Function ID
FUNCTION_NAME	VARCHAR2(18)	Function name
SMALL_READ_MBPS	NUMBER	Single block megabytes read per second
SMALL_WRITE_MBPS	NUMBER	Single block megabytes written per second
LARGE_READ_MBPS	NUMBER	Multiblock megabytes read per second
LARGE_WRITE_MBPS	NUMBER	Multiblock megabytes written per second
SMALL_READ_IOPS	NUMBER	Single block read requests per second
SMALL_WRITE_IOPS	NUMBER	Single block write requests per second
LARGE_READ_IOPS	NUMBER	Multiblock read requests per second
LARGE_WRITE_IOPS	NUMBER	Multiblock write requests per second
AVG_WAIT_TIME	NUMBER	Average wait time (in milliseconds)

V$IOFUNCMETRIC_HISTORY

V$IOFUNCMETRIC_HISTORY displays a recent history of the I/O statistics information by database function for the most recent time interval period.

The columns for V$IOFUNCMETRIC_HISTORY are the same as those for V$IOFUNCMETRIC.

See Also: "V$IOFUNCMETRIC" on page 7-5

V$IOSTAT_CONSUMER_GROUP

V$IOSTAT_CONSUMER_GROUP displays disk I/O statistics for consumer groups. If the resource manager is enabled, then I/O statistics for all consumer groups that are part of the currently enabled resource plan are captured.

Column	Datatype	Description
CONSUMER_GROUP_ID	NUMBER	Consumer Group ID
SMALL_READ_MEGABYTES	NUMBER	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	Number of multiblock megabytes written
SMALL_READ_REQS	NUMBER	Number of single block read requests
SMALL_WRITE_REQS	NUMBER	Number of single block write requests
LARGE_READ_REQS	NUMBER	Number of multiblock read requests
LARGE_WRITE_REQS	NUMBER	Number of multiblock write requests
NUMBER_OF_WAITS	NUMBER	Number of I/O waits by consumer group
WAIT_TIME	NUMBER	Total wait time (in milliseconds)

V$IOSTAT_FILE

V$IOSTAT_FILE displays information about disk I/O statistics of database files (including data files, temp files, and other types of database files). I/O statistics for Data files and Temp files are provided for each file. All other file types (for example, control files, log files, archive logs, and so on) have their statistics consolidated into one entry in the view.

Column	Datatype	Description
FILE_NO	NUMBER	File identification number
FILETYPE_ID	VARCHAR2(25)	Type of file (for example, log file, data file, and so on)
FILETYPE_NAME	VARCHAR2(28)	Name of the file, in the case of a data file or temp file. For all other files, a corresponding string to be displayed (for example, ARCHIVELOG).
SMALL_READ_MEGABYTES	NUMBER	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	Number of multiblock megabytes written
SMALL_READ_REQS	NUMBER	Number of single block read requests
SMALL_WRITE_REQS	NUMBER	Number of single block write requests
SMALL_SYNC_READ_REQS	NUMBER	Number of synchronous single block read requests
LARGE_READ_REQS	NUMBER	Number of multiblock read requests
LARGE_WRITE_REQS	NUMBER	Number of multiblock write requests
SMALL_READ_SERVICETIME	NUMBER	Total service time (in milliseconds) for single block read requests

Column	Datatype	Description
SMALL_WRITE_SERVICETIME	NUMBER	Total service time (in milliseconds) for single block write requests
SMALL_SYNC_READ_LATENCY	NUMBER	Latency for single block synchronous reads (in milliseconds)
LARGE_READ_SERVICETIME	NUMBER	Total service time (in milliseconds) for multiblock read requests
LARGE_WRITE_SERVICETIME	NUMBER	Total service time (in milliseconds) for multiblock write requests
ASYNCH_IO	VARCHAR2(9)	Indicates whether asynchronous I/O is available for the file (ASYNC_ON) or not (ASYNC_OFF)
RETRIES_ON_ERROR	NUMBER	Number of read retries on error

V$IOSTAT_FUNCTION

V$IOSTAT_FUNCTION displays disk I/O statistics for database functions (such as the LGWR and DBWR).

Column	Datatype	Description
FUNCTION_ID	NUMBER	Function ID
FUNCTION_NAME	VARCHAR2(18)	Function name
SMALL_READ_MEGABYTES	NUMBER	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	Number of multiblock megabytes written
SMALL_READ_REQS	NUMBER	Number of single block read requests
SMALL_WRITE_REQS	NUMBER	Number of single block write requests
LARGE_READ_REQS	NUMBER	Number of multiblock read requests
LARGE_WRITE_REQS	NUMBER	Number of multiblock write requests
NUMBER_OF_WAITS	NUMBER	Number of I/O waits by functionality
WAIT_TIME	NUMBER	Total wait time (in milliseconds)

V$IOSTAT_NETWORK

V$IOSTAT_NETWORK displays information about network I/O statistics that were caused by accessing files on a remote database instance.

Column	Datatype	Description
CLIENT	VARCHAR2(32)	Database client name initiating the network I/O (for example, RMAN or PL/SQL)
READS#	NUMBER	Number of read operations issued
WRITES#	NUMBER	Number of write operations issued
KBYTES_READ	NUMBER	Total number of kilobytes read
KBYTES_WRITTEN	NUMBER	Total number of kilobytes written
READ_LATENCY	NUMBER	Total read wait time (in milliseconds)
WRITE_LATENCY	NUMBER	Total write wait time (in milliseconds)

V$JAVA_LIBRARY_CACHE_MEMORY

V$JAVA_LIBRARY_CACHE_MEMORY displays information about memory allocated to library cache memory objects in different namespaces for Java objects. A memory object is an internal grouping of memory for efficient management. A library cache object may consist of one or more memory objects.

Column	Datatype	Description
LC_NAMESPACE	VARCHAR2(15)	Library cache namespace
LC_INUSE_MEMORY_OBJECTS	NUMBER	Number of library cache memory objects currently in use in the Java pool
LC_INUSE_MEMORY_SIZE	NUMBER	Total size of library cache in-use memory objects (in megabytes)
LC_FREEABLE_MEMORY_OBJECTS	NUMBER	Number of freeable library cache memory objects in the Java pool
LC_FREEABLE_MEMORY_SIZE	NUMBER	Size of library cache freeable memory objects (in megabytes)

V$JAVA_POOL_ADVICE

V$JAVA_POOL_ADVICE displays information about estimated parse time in the Java pool for different pool sizes. The sizes range from 10% of the current Java pool size or the amount of pinned Java library cache memory (whichever is higher) to 200% of the current Java pool size, in equal intervals. The value of the interval depends on the current size of the Java pool.

Parse time saved refers to the amount of time saved by keeping library cache memory objects in the Java pool, as opposed to having to reload these objects.

Column	Datatype	Description
JAVA_POOL_SIZE_FOR_ESTIMATE	NUMBER	Java pool size for the estimate (in megabytes)
JAVA_POOL_SIZE_FACTOR	NUMBER	Size factor with respect to the current Java pool size
ESTD_LC_SIZE	NUMBER	Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER	Estimated number of library cache memory objects in the Java pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER	Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a Java pool of the specified size. This is the time that would have been spent in reloading the required objects in the Java pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER	Estimated parse time saved factor with respect to the current Java pool size
ESTD_LC_LOAD_TIME	NUMBER	Estimated elapsed time (in seconds) for parsing in a Java pool of the specified size
ESTD_LC_LOAD_TIME_FACTOR	NUMBER	Estimated load time factor with respect to the current Java pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER	Estimated number of times a library cache memory object was found in a Java pool of the specified size

V$LATCH

V$LATCH displays aggregate latch statistics for both parent and child latches, grouped by latch name. Individual parent and child latch statistics are broken down in the views V$LATCH_PARENT and V$LATCH_CHILDREN.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the latch object
LATCH#	NUMBER	Latch number
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
HASH	NUMBER	Latch hash
GETS	NUMBER	Number of times the latch was requested in willing-to-wait mode
MISSES	NUMBER	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER	Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER	Number of times a no-wait latch request did not succeed (that is, missed)
WAITERS_WOKEN	NUMBER	This column has been deprecated and is present only for compatibility with previous releases of Oracle. No data is accumulated for this column; it will always have a value of zero.
WAITS_HOLDING_LATCH	NUMBER	This column has been deprecated and is present only for compatibility with previous releases of Oracle. No data is accumulated for this column; it will always have a value of zero.
SPIN_GETS	NUMBER	Willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP[1 | 2 | 3 | 4]	NUMBER	These columns have been deprecated and are present only for compatibility with previous releases of Oracle. No data is accumulated for these columns; they will always have a value of zero. As a substitute for this column you can query the appropriate rows of the V$EVENT_HISTOGRAM view where the EVENT column has a value of latch free or latch:%.
SLEEP[5 | 6 | 7 | 8 | 9 | 10 | 11]	NUMBER	These columns have been deprecated and are present only for compatibility with previous releases of Oracle. No data is accumulated for these columns.
WAIT_TIME	NUMBER	Elapsed time spent waiting for the latch (in microseconds)

See Also:

- "V$LATCH_CHILDREN" on page 7-9
- "V$LATCH_PARENT" on page 7-10

V$LATCH_CHILDREN

V$LATCH_CHILDREN displays statistics about child latches. This view includes all columns of V$LATCH plus the CHILD# column. Note that child latches have the same parent if their LATCH# columns match each other.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the latch object
LATCH#	NUMBER	Latch number of the parent latch
CHILD#	NUMBER	Child latch number (unique only to each parent latch)
LEVEL#	NUMBER	Latch level
NAME	VARCHAR2(64)	Latch name
HASH	NUMBER	Latch hash
GETS	NUMBER	Number of times the latch was requested in willing-to-wait mode

Column	Datatype	Description
MISSES	NUMBER	Number of times the latch was requested in willing-to-wait mode and the requestor had to wait
SLEEPS	NUMBER	Number of times a willing-to-wait latch request resulted in a session sleeping while waiting for the latch
IMMEDIATE_GETS	NUMBER	Number of times a latch was requested in no-wait mode
IMMEDIATE_MISSES	NUMBER	Number of times a no-wait latch request did not succeed (that is, missed)
WAITERS_WOKEN	NUMBER	This column has been deprecated and is present only for compatibility with previous releases of Oracle. No data is accumulated for this column; it will always have a value of zero.
WAITS_HOLDING_LATCH	NUMBER	This column has been deprecated and is present only for compatibility with previous releases of Oracle. No data is accumulated for this column; it will always have a value of zero.
SPIN_GETS	NUMBER	Willing-to-wait latch requests which missed the first try but succeeded while spinning
SLEEP[1 | 2 | 3 | 4]	NUMBER	These columns have been deprecated and are present only for compatibility with previous releases of Oracle. No data is accumulated for these columns; they will always have a value of zero. As a substitute for these columns you can query the appropriate rows of the V$EVENT_HISTOGRAM view where the EVENT column has a value of latch free or latch:%.
SLEEP[5 | 6 | 7 | 8 | 9 | 10 | 11]	NUMBER	These columns have been deprecated and are present only for compatibility with previous releases of Oracle. No data is accumulated for these columns.
WAIT_TIME	NUMBER	Elapsed time spent waiting for the latch (in microseconds)

See Also: "V$LATCH" on page 7-8

V$LATCH_MISSES

V$LATCH_MISSES displays statistics about missed attempts to acquire a latch.

Column	Datatype	Description
PARENT_NAME	VARCHAR2(64)	Latch name of a parent latch
WHERE	VARCHAR2(80)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in LOCATION.
NWFAIL_COUNT	NUMBER	Number of times that no-wait acquisition of the latch failed
SLEEP_COUNT	NUMBER	Number of times that acquisition attempts caused sleeps
WTR_SLP_COUNT	NUMBER	Number of times a waiter slept
LONGHOLD_COUNT	NUMBER	Number of times someone held a latch for the entire duration of someone else's sleep
LOCATION	VARCHAR2(80)	Location that attempted to acquire the latch

V$LATCH_PARENT

V$LATCH_PARENT displays statistics about parent latches. The columns of V$LATCH_PARENT are the same as those for V$LATCH.

See Also: "V$LATCH" on page 7-8

V$LATCHHOLDER

V$LATCHHOLDER displays information about the current latch holders.

Column	Datatype	Description
PID	NUMBER	Identifier of the process holding the latch
SID	NUMBER	Identifier of the session that owns the latch
LADDR	RAW(4 | 8)	Latch address
NAME	VARCHAR2(64)	Name of the latch being held
GETS	NUMBER	Number of times that the latch was obtained in either wait mode or no-wait mode

V$LATCHNAME

V$LATCHNAME displays information about decoded latch names for the latches shown in V$LATCH. The rows of V$LATCHNAME have a one-to-one correspondence to the rows of V$LATCH.

Column	Datatype	Description
LATCH#	NUMBER	Latch number
NAME	VARCHAR2(64)	Latch name
HASH	NUMBER	Latch hash

See Also: "V$LATCH" on page 7-8

V$LIBRARY_CACHE_MEMORY

V$LIBRARY_CACHE_MEMORY displays information about memory allocated to library cache memory objects in different namespaces. A memory object is an internal grouping of memory for efficient management. A library cache object may consist of one or more memory objects.

Column	Datatype	Description
LC_NAMESPACE	VARCHAR2(15)	Library cache namespace
LC_INUSE_MEMORY_OBJECTS	NUMBER	Number of library cache memory objects currently in use in the shared pool
LC_INUSE_MEMORY_SIZE	NUMBER	Total size of library cache in-use memory objects (in megabytes)
LC_FREEABLE_MEMORY_OBJECTS	NUMBER	Number of freeable library cache memory objects in the shared pool
LC_FREEABLE_MEMORY_SIZE	NUMBER	Size of library cache freeable memory objects (in megabytes)

V$LIBRARYCACHE

V$LIBRARYCACHE contains statistics about library cache performance and activity.

Column	Datatype	Description
NAMESPACE	VARCHAR2(15)	Library cache namespace
GETS	NUMBER	Number of times a lock was requested for objects of this namespace
GETHITS	NUMBER	Number of times an object's handle was found in memory
GETHITRATIO	NUMBER	Ratio of GETHITS to GETS
PINS	NUMBER	Number of times a PIN was requested for objects of this namespace

Column	Datatype	Description
PINHITS	NUMBER	Number of times all of the metadata pieces of the library object were found in memory
PINHITRATIO	NUMBER	Ratio of PINHITS to PINS
RELOADS	NUMBER	Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from disk
INVALIDATIONS	NUMBER	Total number of times objects in this namespace were marked invalid because a dependent object was modified
DLM_LOCK_REQUESTS	NUMBER	Number of GET requests lock instance locks
DLM_PIN_REQUESTS	NUMBER	Number of PIN requests lock instance locks
DLM_PIN_RELEASES	NUMBER	Number of release requests PIN instance locks
DLM_INVALIDATION_REQUESTS	NUMBER	Number of GET requests for invalidation instance locks
DLM_INVALIDATIONS	NUMBER	Number of invalidation pings received from other instances

V$LICENSE

V$LICENSE displays information about license limits.

Column	Datatype	Description
SESSIONS_MAX	NUMBER	Maximum number of concurrent user sessions allowed for the instance
SESSIONS_WARNING	NUMBER	Warning limit for concurrent user sessions for the instance
SESSIONS_CURRENT	NUMBER	Current number of concurrent user sessions
SESSIONS_HIGHWATER	NUMBER	Highest number of concurrent user sessions since the instance started
USERS_MAX	NUMBER	Maximum number of named users allowed for the database
CPU_COUNT_CURRENT	NUMBER	Current number of logical CPUs or processors on the system
CPU_CORE_COUNT_CURRENT	NUMBER	Current number of CPU cores on the system (includes subcores of multicore CPUs, as well as single-core CPUs)
CPU_SOCKET_COUNT_CURRENT	NUMBER	Current number of CPU sockets on the system (represents an absolute count of CPU chips on the system, regardless of multithreading or multicore architectures)
CPU_COUNT_HIGHWATER	NUMBER	Highest number of logical CPUs or processors on the system since the instance started
CPU_CORE_COUNT_HIGHWATER	NUMBER	Highest number of CPU cores on the system since the instance started (includes subcores of multicore CPUs, as well as single-core CPUs)
CPU_SOCKET_COUNT_HIGHWATER	NUMBER	Highest number of CPU sockets on the system since the instance started (represents an absolute count of CPU chips on the system, regardless of multithreading or multicore architectures)

Note: The availability of the CPU core count and CPU socket count statistics is subject to the operating system platform on which the Oracle Database is running. If a statistic is unavailable, the view will return NULL for the statistic value.

V$LOADISTAT

V$LOADISTAT contains errors that occurred when updating indexes on a table during a load using the Direct Path API.

Column	Datatype	Description
OWNER	VARCHAR2(31)	Schema name
TABNAME	VARCHAR2(31)	Table name
INDEXNAME	VARCHAR2(31)	Index name
SUBNAME	VARCHAR2(31)	Index sub name
MESSAGE_NUM	NUMBER	Error message number
MESSAGE	VARCHAR2(4000)	Error message

V$LOADPSTAT

V$LOADPSTAT contains statistics about the number of rows loaded into a partition, or subpartition, during a load using the Direct Path API.

Column	Datatype	Description
OWNER	VARCHAR2(31)	Schema name
TABNAME	VARCHAR2(31)	Table name
PARTNAME	VARCHAR2(31)	Partition name
LOADED	NUMBER	Number of rows loaded

V$LOCK

V$LOCK lists the locks currently held by the Oracle Database and outstanding requests for a lock or latch.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of lock state object
KADDR	RAW(4 | 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of user or system lock The locks on the user types are obtained by user applications. Any process that is blocking others is likely to be holding one of these locks. The user type locks are: TM - DML enqueue TX - Transaction enqueue UL - User supplied The locks on the system types are held for extremely short periods of time. The system type locks are listed in Table 7–1.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)

Column	Datatype	Description
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	A value of either 0 or 1, depending on whether or not the lock in question is the blocker.

Table 7–1 Values for the TYPE Column: System Types

System Type	Description	System Type	Description
BL	Buffer hash table instance	NA..NZ	Library cache pin instance (A..Z = namespace)
CF	Control file schema global enqueue	PF	Password File
CI	Cross-instance function invocation instance	PI, PS	Parallel operation
CU	Cursor bind	PR	Process startup
DF	datafile instance	QA..QZ	Row cache instance (A..Z = cache)
DL	Direct loader parallel index create	RT	Redo thread global enqueue
DM	Mount/startup db primary/secondary instance	SC	System change number instance
DR	Distributed recovery process	SM	SMON
DX	Distributed transaction entry	SN	Sequence number instance
FS	File set	SQ	Sequence number enqueue
HW	Space management operations on a specific segment	SS	Sort segment
IN	Instance number	ST	Space transaction enqueue
IR	Instance recovery serialization global enqueue	SV	Sequence number value
IS	Instance state	TA	Generic enqueue
IV	Library cache invalidation instance	TS	Temporary segment enqueue (ID2=0)
JQ	Job queue	TS	New block allocation enqueue (ID2=1)
KK	Thread kick	TT	Temporary table enqueue
LA .. LP	Library cache lock instance lock (A..P = namespace)	UN	User name
MM	Mount definition global enqueue	US	Undo segment DDL
MR	Media recovery	WL	Being-written redo log instance

V$LOCK_ACTIVITY

V$LOCK_ACTIVITY is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
FROM_VAL	CHAR(4)	Global Cache Resource initial state; always NULL

Column	Datatype	Description
TO_VAL	CHAR(1)	Global Cache Resource initial state; always S
ACTION_VAL	CHAR(21)	Description of the conversion; always Lock buffers for read
COUNTER	NUMBER	Number of times the lock operation executed

V$LOCK_TYPE

V$LOCK_TYPE ???.

Column	Datatype	Description
TYPE	VARCHAR2(64)	A two-letter internal resource identifier
NAME	VARCHAR2(64)	Resource type name. This is a short (less than 32 characters) enqueue type name.
ID1_TAG	VARCHAR2(64)	Description of the enqueue type.
ID2_TAG	VARCHAR2(64)	Further description of the enqueue type.
IS_USER	VARCHAR2(3)	User enqueue. These are enqueues that are acquired as a direct result of a SQL statement. Applications may get deadlocks on these enqueues. Such deadlocks are considered application errors.
DESCRIPTION	VARCHAR2(4000)	Explanation of how or for what purpose the enqueue is used.

V$LOCKED_OBJECT

V$LOCKED_OBJECT lists all locks acquired by every transaction on the system. It shows which sessions are holding DML locks (that is, TM-type enqueues) on what objects and in what mode.

Column	Datatype	Description
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
OBJECT_ID	NUMBER	Object ID being locked
SESSION_ID	NUMBER	Session ID
ORACLE_USERNAME	VARCHAR2(30)	Oracle user name
OS_USER_NAME	VARCHAR2(30)	OS user name
PROCESS	VARCHAR2(12)	OS process ID
LOCKED_MODE	NUMBER	Lock mode

V$LOG

V$LOG displays log file information from the control file.

Column	Datatype	Description
GROUP#	NUMBER	Log group number
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
BYTES	NUMBER	Size of the log (in bytes)
MEMBERS	NUMBER	Number of members in the log group
ARCHIVED	VARCHAR2(3)	Archive status (YES) or (NO)

Column	Datatype	Description
STATUS	VARCHAR2(16)	Log status: ■ UNUSED - Online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. ■ CURRENT - Current redo log. This implies that the redo log is active. The redo log could be open or closed. ■ ACTIVE - Log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It may or may not be archived. ■ CLEARING - Log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE statement. After the log is cleared, the status changes to UNUSED. ■ CLEARING_CURRENT - Current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. ■ INACTIVE - Log is no longer needed for instance recovery. It may be in use for media recovery. It may or may not be archived.
FIRST_CHANGE#	NUMBER	Lowest system change number (SCN) in the log
FIRST_TIME	DATE	Time of the first SCN in the log

V$LOG_HISTORY

V$LOG_HISTORY displays log history information from the control file.

Column	Datatype	Description
RECID	NUMBER	Control file record ID
STAMP	NUMBER	Control file record stamp
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
FIRST_CHANGE#	NUMBER	Lowest system change number (SCN) in the log
FIRST_TIME	DATE	Time of the first entry (lowest SCN) in the log
NEXT_CHANGE#	NUMBER	Highest SCN in the log
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when the log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when the log was written

V$LOGFILE

V$LOGFILE contains information about redo log files.

Column	Datatype	Description
GROUP#	NUMBER	Redo log group identifier number
STATUS	VARCHAR2(7)	Status of the log member: ■ INVALID - File is inaccessible ■ STALE - File's contents are incomplete ■ DELETED - File is no longer used ■ null - File is in use
TYPE	VARCHAR2(7)	Type of the logfile: ■ ONLINE ■ STANDBY
MEMBER	VARCHAR2(513)	Redo log member name

Column	Datatype	Description
IS_RECOVERY_DEST_FILE	VARCHAR2(3)	Indicates whether the file was created in the flash recovery area (YES) or not (NO)

V$LOGHIST

V$LOGHIST contains log history information from the control file. This view is retained for historical compatibility. Oracle recommends that you use V$LOG_HISTORY instead.

See Also: "V$LOG_HISTORY" on page 7-16

Column	Datatype	Description
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of first SCN in the log
SWITCH_CHANGE#	NUMBER	SCN at which the log switch occurred; one more than highest SCN in the log

V$LOGMNR_CONTENTS

V$LOGMNR_CONTENTS contains log history information. To query this view, you must have the SELECT ANY TRANSACTION privilege.

When a SELECT statement is executed against the V$LOGMNR_CONTENTS view, the archive redo log files are read sequentially. Translated records from the redo log files are returned as rows in the V$LOGMNR_CONTENTS view. This continues until either the filter criteria specified at startup (EndTime or endScn) are met or the end of the archive log file is reached.

Column	Datatype	Description
SCN	NUMBER	System change number (SCN) when the database change was made
START_SCN	NUMBER	System change number (SCN) when the transaction that contains this change started; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation, NULL otherwise. This column may also be NULL if the query is run over a time/SCN range that does not contain the start of the transaction.
COMMIT_SCN	NUMBER	System change number (SCN) when the transaction committed; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation
TIMESTAMP	DATE	Timestamp when the database change was made
START_TIMESTAMP	DATE	Timestamp when the transaction that contains this change started; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation, NULL otherwise. This column may also be NULL if the query is run over a time/SCN range that does not contain the start of the transaction.
COMMIT_TIMESTAMP	DATE	Timestamp when the transaction committed; only meaningful if the COMMITTED_DATA_ONLY option was chosen in a DBMS_LOGMNR.START_LOGMNR() invocation
XIDUSN	NUMBER	Transaction ID undo segment number of the transaction that generated the change
XIDSLT	NUMBER	Transaction ID slot number of the transaction that generated the change

Column	Datatype	Description
XIDSQN	NUMBER	Transaction ID sequence number of the transaction that generated the change
XID	RAW(8)	Raw representation of the transaction identifier
PXIDUSN	NUMBER	Parent transaction ID undo segment number of a parallel transaction
PXIDSLT	NUMBER	Parent transaction ID slot number of a parallel transaction
PXIDSQN	NUMBER	Parent transaction ID sequence number of a parallel transaction
PXID	RAW(8)	Raw representation of the parent transaction identifier
TX_NAME	VARCHAR2(256)	Name of the transaction that made the change; only meaningful if the transaction is a named transaction
OPERATION	VARCHAR2(32)	User level SQL operation that made the change: ■ INSERT - Change was caused by an insert statement ■ UPDATE - Change was caused by an update statement ■ DELETE - Change was caused by a delete statement ■ DDL - Change was caused by a DDL statement ■ START - Change was caused by the start of a transaction ■ COMMIT - Change was caused by the commit of a transaction ■ ROLLBACK - Change was caused by a full rollback of a transaction ■ LOB_WRITE - Change was caused by an invocation of DBMS_LOB.WRITE ■ LOB_TRIM - Change was caused by an invocation of DBMS_LOB.TRIM ■ LOB_ERASE - Change was caused by an invocation of DBMS_LOB.ERASE ■ SELECT_FOR_UPDATE - Operation was a SELECT FOR UPDATE statement ■ SEL_LOB_LOCATOR - Operation was a SELECT statement that returned a LOB locator ■ MISSING_SCN - LogMiner encountered a gap in the redo records. This is most likely because not all redo logs were registered with LogMiner. ■ INTERNAL - Change was caused by internal operations initiated by the database ■ XML DOC BEGIN - Beginning of a change to an XMLType column or table ■ XML DOC WRITE - Data for an XML document ■ XML DOC END - End of the Data for an XML document ■ UNSUPPORTED - Change was caused by operations not currently supported by LogMiner (for example, changes made to tables with ADT columns)

Column	Datatype	Description
OPERATION_CODE	NUMBER	Number of the operation code: ■ 0 - INTERNAL ■ 1 - INSERT ■ 2 - DELETE ■ 3 - UPDATE ■ 5 - DDL ■ 6 - START ■ 7 - COMMIT ■ 9 - SELECT_LOB_LOCATOR ■ 10 - LOB_WRITE ■ 11 - LOB_TRIM ■ 25 - SELECT_FOR_UPDATE ■ 28 - LOB_ERASE ■ 34 - MISSING_SCN ■ 68 - XML DOC BEGIN ■ 70 = XML DOC WRITE ■ 71 = XML DOC END ■ 36 - ROLLBACK ■ 255 - UNSUPPORTED
ROLLBACK	NUMBER	1 = if the redo record was generated because of a partial or a full rollback of the associated transaction 0 = otherwise
SEG_OWNER	VARCHAR2(32)	Owner of the modified data segment
SEG_NAME	VARCHAR2(256)	Name of the modified data segment
TABLE_NAME	VARCHAR2(32)	Name of the modified table (in case the redo pertains to a table modification)
SEG_TYPE	NUMBER	Type of the modified data segment: ■ 0 - UNKNOWN ■ 1 - INDEX ■ 2 - TABLE ■ 19 - TABLE PARTITION ■ 20 - INDEX PARTITION ■ 34 - TABLE SUBPARTITION ■ All other values - UNSUPPORTED
SEG_TYPE_NAME	VARCHAR2(32)	Segment type name: ■ UNKNOWN ■ INDEX ■ TABLE ■ TABLE PARTITION ■ INDEX PARTITION ■ TABLE SUBPARTITION ■ UNSUPPORTED
TABLE_SPACE	VARCHAR2(32)	Name of the tablespace containing the modified data segment. This column is not populated for rows where the value of the OPERATION column is DDL. This is because DDL may operate on more than one tablespace.
ROW_ID	VARCHAR2(18)	Row ID of the row modified by the change (only meaningful if the change pertains to a DML). This will be NULL if the redo record is not associated with a DML.
USERNAME	VARCHAR2(30)	Name of the user who executed the transaction

Column	Datatype	Description
OS_USERNAME	VARCHAR2(4000)	Name of the operating system user
MACHINE_NAME	VARCHAR2(4000)	Machine from which the user connected to the database
AUDIT_SESSIONID	NUMBER	Audit session ID associated with the user session making the change
SESSION#	NUMBER	Session number of the session that made the change
SERIAL#	NUMBER	Serial number of the session that made the change
SESSION_INFO	VARCHAR2(4000)	Information about the database session that executed the transaction. Contains process information, machine name from which the user logged in, and so on. A possible SESSION_INFO column may contain the following: ■ login_username = HR ■ client_info = ■ OS_username = jkundu ■ Machine_name = nirvan ■ OS_terminal = pts/31 ■ OS_program_name = sqlplus@nirvan (TNS V1-V3)
THREAD#	NUMBER	Number of the thread that made the change to the database
SEQUENCE#	NUMBER	Sequence number of the redo log that contained the redo record corresponding to the database change
RBASQN	NUMBER	Sequence# associated with the Redo Block Address (RBA) of the redo record associated with the change
RBABLK	NUMBER	RBA block number within the log file
RBABYTE	NUMBER	RBA byte offset within the block
UBAFIL	NUMBER	Undo Block Address (UBA) file number identifying the file containing the undo block
UBABLK	NUMBER	UBA block number for the undo block
UBAREC	NUMBER	UBA record index within the undo block
UBASQN	NUMBER	UBA undo block sequence number
ABS_FILE#	NUMBER	Data block absolute file number of the block changed by the transaction
REL_FILE#	NUMBER	Data block relative file number. The file number is relative to the tablespace of the object.
DATA_BLK#	NUMBER	Data block number within the file
DATA_OBJ#	NUMBER	Data block object number identifying the object
DATA_OBJV#	NUMBER	Version number of the table being modified
DATA_OBJD#	NUMBER	Data block data object number identifying the object within the tablespace
SQL_REDO	VARCHAR2(4000)	Reconstructed SQL statement that is equivalent to the original SQL statement that made the change. Refer to *Oracle Database Utilities* before executing SQL_REDO to your database. LogMiner does not generate SQL redo for temporary tables. In such a case, this column will contain the string "/* No SQL_REDO for temporary tables */".
SQL_UNDO	VARCHAR2(4000)	Reconstructed SQL statement that can be used to undo the effect of the original statement that made the change. DDL statements have no corresponding SQL_UNDO. Refer to *Oracle Database Utilities* before executing SQL_UNDO to your database. LogMiner does not generate SQL undo for temporary tables. In such a case, this column will contain the string "/* No SQL_UNDO for temporary tables */".

Column	Datatype	Description
RS_ID	VARCHAR2(32)	Record set ID. The tuple (RS_ID, SSN) together uniquely identifies a logical row change. This will usually mean one row from V$LOGMNR_CONTENTS, but could be more than one row if a single SQL statement for either the Redo or Undo would be too large to fit within the respective columns SQL_UNDO or SQL_REDO. RS_ID uniquely identifies the redo record that generated the row.
SSN	NUMBER	SQL sequence number. Used in conjunction with RS_ID, this uniquely identifies a logical row change, shown as one or more rows from the V$LOGMNR_CONTENTS view.
CSF	NUMBER	Continuation SQL flag. Possible values are: ▪ 0 - Indicates SQL_REDO and SQL_UNDO is contained within the same row ▪ 1 - Indicates that either SQL_REDO or SQL_UNDO is greater than 4000 bytes in size and is continued in the next row returned by the view
INFO	VARCHAR2(32)	Informational message about the row. For instance, the string "USER DDL" indicates that the DDL statement returned in the SQL_REDO column was the top-level DDL executed by the user and the string "INTERNAL DDL" indicates that the DDL statement returned in the SQL_REDO column was executed internally by the RDBMS.
STATUS	NUMBER	A value of 0 indicates that the reconstructed SQL statements as shown in the SQL_REDO and SQL_UNDO columns are valid executable SQL statements. Otherwise, the reconstructed SQL statements are not executable. This may be due to the fact that no data dictionary was provided to LogMiner for the analysis, or that the data dictionary provided did not have the definition of the object being mined. A value of 5 indicates that this row is part of a change to an XMLType column or table and the XML document must be assembled before being applied.
REDO_VALUE	NUMBER	Used as input to the DBMS_LOGMNR.MINE_VALUE() and DBMS_LOGMNR.COLUMN_PRESENT() functions
UNDO_VALUE	NUMBER	Used as input to the DBMS_LOGMNR.MINE_VALUE() and DBMS_LOGMNR.COLUMN_PRESENT() functions
SAFE_RESUME_SCN	NUMBER	Reserved for future use
CSCN	NUMBER	This column is deprecated in favor of the COMMIT_SCN column
OBJECT_ID	RAW(16)	Object identifier for DMLs to XMLType tables. For changes to non-typed tables, this column is NULL.

V$LOGMNR_DICTIONARY

V$LOGMNR_DICTIONARY contains log history information.

Column	Datatype	Description
DB_NAME	VARCHAR2(9)	Name of the database
DB_ID	NUMBER	Database ID
DB_CREATED	DATE	Creation date of the source database (corresponds to the CREATED column in the V$DATABASE view)
TIMESTAMP	DATE	Date when the dictionary was created
RESET_SCN	NUMBER	Reset log SCN when the dictionary was created
RESET_SCN_TIME	DATE	Timestamp of the reset log SCN when the dictionary was created
DB_VERSION_TIME	DATE	Version time for the source database (corresponds to the VERSION_TIME column in the V$DATABASE view)
DB_CHARACTER_SET	VARCHAR2(30)	Character set of the source database
DB_VERSION	VARCHAR2(64)	This column is deprecated.
DB_STATUS	VARCHAR2(64)	This column is deprecated.

Column	Datatype	Description
DICTIONARY_SCN	NUMBER	Database checkpoint SCN at which the dictionary was created
ENABLED_THREAD_MAP	RAW(16)	This column is deprecated.
DB_TXN_SCN	NUMBER	SCN at which the dictionary was created
FILENAME	VARCHAR2(512)	Dictionary file name
INFO	VARCHAR2(32)	Informational/Status message BAD_DATE indicates that the SCN of the dictionary file does not match the SCN range of the log files
STATUS	NUMBER	A NULL indicates a valid dictionary file for the list of log files. A non-NULL value indicates further information is contained in the INFO column as a text string.

V$LOGMNR_DICTIONARY_LOAD

V$LOGMNR_DICTIONARY_LOAD displays information about LogMiner dictionaries for all active LogMiner sessions on the system. Each query of this view will return one row for each attached LogMiner session. This view will not show valid information for LogMiner adhoc query clients.

Column	Datatype	Description
SESSION_ID	NUMBER	LogMiner session ID
LOGMNR_UID	NUMBER	LogMiner dictionary UID
ACTION#	NUMBER	For internal use only
OPCODE	NUMBER	For internal use only
COMMAND	VARCHAR2(161)	Current command being executed
CURRENT_STATE	VARCHAR2(32)	Summary information if LOADED=ACTIVE
COMPLETED_ACTIONS	NUMBER	The steps completed so far
TOTAL_ACTIONS	NUMBER	Total steps to complete
LOADED	VARCHAR2(7)	Status of the dictionary load: ■ YES - dictionary has been loaded ■ NO - dictionary has not been loaded ■ ACTIVE - dictionary is in the process of being loaded
PERCENT_DONE	NUMBER	Percentage of the dictionary that has been loaded

V$LOGMNR_LATCH

V$LOGMNR_LATCH can be joined with the V$LATCH and the V$LATCH_CHILDREN views to obtain statistics about different latches used by active LogMiner persistent sessions. A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating a Streams capture.

Column	Datatype	Description
SESSION_ID	NUMBER	Unique identifier of the LogMiner persistent session

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the Latch: - LogMiner internal state - Identifies the latch that protects communications between the LogMiner READER, PREPARER, and BUILDER processes - LogMiner memory allocation - Identifies the latch that protects all memory allocation and deallocation inside LogMiner - LogMiner transaction list - Identifies the latch that protects interaction between the LogMiner layer and its clients (Data Guard SQL Apply on a logical standby database or Oracle Streams)
CHILD_ADDR	RAW(4)	Address of the child latch object. This column matches the corresponding ADDR column in the V$LATCH_CHLDREN view
STATE	VARCHAR2(6)	State of the Latch: - UNINIT - The latch structure is uninitialized; that is, it is not currently assigned to any LogMiner persistent session - READY - The latch structure is being used by a persistent LogMiner session

V$LOGMNR_LOGS

V$LOGMNR_LOGS contains log information.

Column	Datatype	Description
LOG_ID	NUMBER	This column is deprecated.
FILENAME	VARCHAR2(512)	Name of the log file
LOW_TIME	DATE	Oldest date of any records in the file
HIGH_TIME	DATE	Most recent date of any records in the file
DB_ID	NUMBER	Database ID
DB_NAME	VARCHAR2(8)	Name of the database
RESET_SCN	NUMBER	Resetlogs SCN of the database incarnation that generated the log file
RESET_SCN_TIME	DATE	Resetlogs timestamp of the database incarnation that generated the log file
THREAD_ID	NUMBER	Thread number
THREAD_SQN	NUMBER	Thread sequence number
LOW_SCN	NUMBER	SCN allocated when log switched into
NEXT_SCN	NUMBER	SCN after this log. Low SCN of the next log.
DICTIONARY_BEGIN	VARCHAR2(3)	Indicates whether dictionary dumped to redo logs starts in this redo log (YES) or not (NO)
DICTIONARY_END	VARCHAR2(3)	Indicates whether dictionary dumped to redo logs ends in this redo log (YES) or not (NO)
TYPE	VARCHAR2(7)	Redo log file type: - ARCHIVED - ONLINE
BLOCKSIZE	NUMBER	Database block size
FILESIZE	NUMBER	Size of the redo file (in bytes)
INFO	VARCHAR2(32)	Informational message. A value of MISSING_LOGFILE will be assigned to a row entry where a needed log file is missing from the list of log files.
STATUS	NUMBER	Status of the redo log file: - 0 - Will be read - 1 - First to be read - 2 - Not needed - 4 - Missing log file

V$LOGMNR_PARAMETERS

V$LOGMNR_PARAMETERS contains log information.

Column	Datatype	Description
START_DATE	DATE	Date to start the search
REQUIRED_START_DATE	DATE	Required date to start the search if DDL tracking is enabled
END_DATE	DATE	Date to end the search
START_SCN	NUMBER	System change number to start the search
REQUIRED_START_SCN	NUMBER	Required system change number to start the search if DDL tracking is enabled
END_SCN	NUMBER	System change number to end the search
OPTIONS	NUMBER	Options specified for the current LogMiner session
INFO	VARCHAR2(32)	This column is always null.
STATUS	NUMBER	This column is always 0.

V$LOGMNR_PROCESS

V$LOGMNR_PROCESS identifies all processes attached to an active LogMiner persistent session. (A persistent LogMiner session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams capture.) This view can be joined with either the V$SESSION view or the V$PROCESS view to gather process-specific information.

Column	Datatype	Description
SESSION_ID	NUMBER	Unique identifier for the LogMiner persistent session
PID	NUMBER	Oracle process identifier for the SQL Apply or Streams capture process (same as the V$PROCESS.PID)
SPID	VARCHAR2(12)	Operating system process identifier (same as the V$PROCESS.SPID)
ROLE	VARCHAR2(32)	Identifies the role of the active LogMiner process: READER, PREPARER, BUILDER, COORDINATOR, or APPLY SERVER
USERNAME	VARCHAR2(15)	Operating system process user name that is connected to the database
SID	NUMBER	Session identifier for the V$SESSION.SID process
SERIAL#	NUMBER	Session serial number associated with the V$SESSION.SERIAL process
LATCHWAIT	VARCHAR2(8)	Address of the latch the process is waiting for; NULL if none
LATCHSPIN	VARCHAR2(8)	Address of the latch the process is spinning on; NULL if none
WORK_MICROSEC	VARCHAR2(21)	Microseconds spent by the process doing useful work
OVERHEAD_MICROSEC	VARCHAR2(21)	Microseconds spent by the process doing overhead tasks or simply waiting/idling

V$LOGMNR_SESSION

V$LOGMNR_SESSION displays information about active LogMiner persistent sessions. (A LogMiner persistent session is created either by starting Data Guard SQL Apply on a logical standby database for the first time or by creating Streams Capture.) Transient LogMiner sessions (those created as a result of querying the V$LOGMNR_CONTENTS view) do not show up in the V$LOGMNR_SESSION view. The statistics shown in this view correspond to snapshots of the system and are not cumulative in nature.

Column	Datatype	Description
SESSION_ID	NUMBER	Unique identifier for the LogMiner persistent session
SESSION_NAME	VARCHAR2(32)	Unique session name
SESSION_STATE	VARCHAR2(9)	Current state of the LogMiner persistent session: ■ READY - Client has created the LogMiner persistent session and added the relevant archived redo log files, but has not loaded the initial LogMiner dictionary ■ STARTED - The LogMiner dictionary has been loaded ■ ACTIVE - The LogMiner persistent session is mining the redo stream ■ DETACHED - The LogMiner persistent session is not currently active, and it is in the process of becoming inactive ■ DISCARDED - Client is about to destroy the LogMiner persistent session
DB_NAME	VARCHAR2(128)	Global database name for the source database
DB_ID	NUMBER	Database identifier of the source database
RESET_SCN	NUMBER	System change number (SCN) when the session started
RESET_TIMESTAMP	NUMBER	Time of the RESETLOGs when the LogMiner persistent session started
NUM_PROCESS	NUMBER	Number of processes allocated to this session
CHUNK_SIZE	NUMBER	Amount of memory allocated for this chunk
START_SCN	NUMBER	System change number (SCN) at start of the session
END_SCN	NUMBER	System change number (SCN) at end of the session
SPILL_SCN	NUMBER	In the event of a restart, redo records with an SCN lower than this will not be read from the archived redo log files
PROCESSED_SCN	NUMBER	The BUILDER process has successfully mined redo records up to this SCN
PREPARED_SCN	NUMBER	The PREPARER processes have successfully transformed all redo records below this SCN into logical change records (LCRs). However the LCRs may not have been grouped into transactions or merged in case they pertain to chained rows or LOB updates.
READ_SCN	NUMBER	The READER process has read all redo records below this SCN
LOW_MARK_SCN	NUMBER	LogMiner has delivered all transactions that committed below this SCN to the client
CONSUMED_SCN	NUMBER	Client has consumed and released all transactions that committed below this SCN.
MAX_MEMORY_SIZE	NUMBER	Maximum amount of shared memory that LogMiner is allowed to consume
USED_MEMORY_SIZE	NUMBER	Amount of shared memory actually consumed by LogMiner
BUILDER_WORK_SIZE	NUMBER	???
PREPARED_WORK_SIZE	NUMBER	Amount of redo that has been prepared by LogMiner
AVAILABLE_WORK_SIZE	NUMBER	Amount of redo records that are ready, but are yet to be consumed by the client
AVAILABLE_TXN	NUMBER	Number of transaction chunks ready for consumption
AVAILABLE_COMMITTED_TXN	NUMBER	Number of committed transactions ready for consumption. This is less than, or equal to, AVAILABLE_TXN.
DELIVERED_TXN	NUMBER	Number of transaction chunks that the client currently has in its possession
DELIVERED_COMMITTED_TXN	NUMBER	Number of committed transaction chunks that the client is currently working on. This is less than, or equal to, the value of the DELIVERED_TXN column.
PINNED_TXN	NUMBER	Number of transactions pinned (the client is actively working on)
PINNED_COMMITTED_TXN	NUMBER	Number of committed transactions pinned (the client is actively working on)

Column	Datatype	Description
CHECKPOINT_INTERVAL	NUMBER	Checkpoint interval

V$LOGMNR_STATS

V$LOGMNR_STATS displays the activity currently being performed by the active LogMiner persistent sessions.

Column	Datatype	Description
SESSION_ID	NUMBER	Unique identifier for the LogMiner persistent sessions
NAME	VARCHAR2(64)	Name of the LogMiner statistic, state, or status, including: ■ Bytes of Redo Processed - Cumulative count of bytes processed by SQL Apply ■ Redo Records Processed - Count of redo records processed by SQL Apply ■ Txns Delivered to Client - Count of SQL transactions processed by SQL Apply ■ DML txns delivered - Count of DML transactions processed by SQL Apply ■ DDL txns delivered - Count of DDL transactions processed by SQL Apply ■ CTAS txns delivered - Count of CREATE TABLE AS SELECT (CTAS) transactions processed by SQL Apply ■ Recursive txns delivered - Count of recursive transactions processed by SQL Apply ■ Rolled back txns seen ■ LCRs delivered to client - Number of logical change records (LCRs) processed by SQL Apply ■ Bytes paged out - Cumulative count of bytes that have been paged out. LogMiner pages out memory from the LCR cache to accommodate certain ill-behaved workloads or under-configured systems. The ratio of bytes paged out to bytes of redo processed should be low. If this ratio is high (10% or higher), try increasing the MAX_SGA allocated to SQL Apply. ■ Microsecs spent in pageout - Time spent by LogMiner paging out memory from the LCR cache ■ Bytes checkpointed - Keeps track of the amount of bytes checkpointed. The mining engine takes periodic checkpoints, whereby it writes out logical change records (LCRs) pertaining to long-running transactions. The ratio of Bytes Checkpointed to Bytes of Redo Processed should be low. A high ratio (10% or higher) indicates an ill-behaved workload. ■ Microsecs spent in checkpoint - Time spent by the mining engine taking checkpoints, whereby it writes out logical change records (LCRs) pertaining to long-running transactions. ■ Bytes rolled back - Cumulative value of the number of bytes rolled back by LogMiner. There are times that LogMiner needs to backtrack and reprocess a section of the redo stream. In this case, it will roll back work it has already done. The ratio of Bytes Rolled Back to Bytes of Redo Processed should be low. If this ratio is high (10% or higher), reduce the number of PREPARER processes allocated to SQL Apply. ■ Microsecs spent in rollback - Time spent rolling back transactions already applied to the logical standby database
VALUE	VARCHAR2(64)	The corresponding metric value

V$LOGSTDBY

V$LOGSTDBY is deprecated.

Column	Datatype	Description
SERIAL#	NUMBER	SQL Session serial number. This data is used when joining this view with the V$SESSION and V$PX_SESSION views.
LOGSTDBY_ID	NUMBER	Parallel query slave ID
PID	VARCHAR2(12)	Process ID of the SQL apply process
TYPE	VARCHAR2(30)	Indicates the task being performed by the process (COORDINATOR, APPLIER, ANALYZER, READER, PREPARER, or BUILDER)
STATUS_CODE	NUMBER	Status number (or Oracle error code) belonging to the STATUS message
STATUS	VARCHAR2(256)	Description of the current activity of the process
HIGH_SCN	NUMBER	Highest system change number (SCN) seen by the process. This column is used to confirm the progress of the individual process.

V$LOGSTDBY_PROCESS

V$LOGSTDBY_PROCESS displays dynamic information about what is happening to the Data Guard log apply services. This view is very helpful when diagnosing performance problems during the logical application of archived redo logs to the standby database, and it can be helpful for other problems. This view is for logical standby databases only.

Column	Datatype	Description
SID	NUMBER	The session id of the associated session. This matches the SID column of the corresponding row in the V$SESSION view.
SERIAL#	NUMBER	The serial number of the associated session. (SID,SERIAL#) together uniquely identifies the session in the current database instance.
LOGSTDBY_ID	NUMBER	Parallel query slave ID
SPID	VARCHAR2(12)	This corresponds to the SPID value of the row corresponding to this process in the V$PROCESS view
TYPE	VARCHAR2(30)	The role that the process plays in the context of SQL Apply. Can take one of the following values: COORDINATOR, APPLIER, ANALYZER, READER, PREPARER, or BUILDER
STATUS_CODE	NUMBER	Operation code identifying the current action of the process
STATUS	VARCHAR2(256)	Description of the current action of the process
HIGH_SCN	NUMBER	Identifies the highest redo record/LCR processed by this process

V$LOGSTDBY_PROGRESS

V$LOGSTDBY_PROGRESS displays the progress of log apply services on the logical standby database. This view is for logical standby databases only.

Column	Datatype	Description
APPLIED_SCN	NUMBER	All the transactions with COMMIT SCN lower than or equal to this SCN have been applied
APPLIED_TIME	DATE	The time and date of APPLIED_SCN
RESTART_SCN	NUMBER	During an apply restart, LogMiner does not read any log file with a NEXT_CHANGE# lower than this SCN.
RESTART_TIME	DATE	The time and date of RESTART_SCN
LATEST_SCN	NUMBER	The highest SCN of all redo records that Logical Standby has encountered
LATEST_TIME	DATE	The time and date of LATEST_SCN
MINING_SCN	NUMBER	The SCN of the latest redo record processed by the builder process

Column	Datatype	Description
MINING_TIME	DATE	The time and date of MINING_SCN

V$LOGSTDBY_STATE

V$LOGSTDBY_STATE provides consolidated information from V$LOGSTDBY and V$LOGSTDBY_STATS about the running state of Logical Standby.

Column	Datatype	Description
PRIMARY_DBID	NUMBER	Database ID (DBID) of the primary database
SESSION_ID	NUMBER	LogMiner session ID allocated to SQL Apply.
REALTIME_APPLY	VARCHAR2(64)	Y indicates that SQL Apply is running in real-time apply mode. If a standby redo log is configured, SQL Apply applies changes as they are written to the standby redo log files. N indicates that SQL Apply applies changes as each archived redo log file is received.
STATE	VARCHAR2(64)	■ INITIALIZING: LogMiner session has been created and coordinator has attached to it ■ LOADING DICTIONARY: SQL Apply is loading the LogMiner dictionary ■ WAITING ON GAP: SQL Apply is waiting for a log file to be sent from the primary database ■ APPLYING: SQL Apply is actively mining or applying transactions ■ WAITING FOR DICTIONARY LOGS: SQL Apply is waiting for the archived logs containing the LogMiner dictionary to be shipped from the primary database ■ IDLE: SQL Apply has applied all changes available at the logical standby, and is caught up with the primary database

V$LOGSTDBY_STATS

V$LOGSTDBY_STATS displays statistics, current state, and status information related to SQL Apply. No rows are returned from this view when SQL Apply is not running. This view is only meaningful in the context of a logical standby database.

All statistics shown in this view are reinitialized at each SQL Apply start.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the statistic, state, or status: **Note:** Many of the following statistics are subject to change or deletion; programmers should write application code to tolerate missing or extra statistics. ■ id of the logminer session used by SQL Apply to mine the redo logs. ■ number of preparers ■ number of appliers ■ server processes in use for SQL Apply ■ maximum SGA (in MBytes) for LCR cache ■ whether SQL Apply is preserving the commit order seen at the primary database while applying changes ■ maximum events recorded in the DBA_LOGSTDBY_EVENTS table ■ whether SQL Apply is logging errors that are skipped in the DBA_LOGSTDBY_EVENTS table ■ whether SQL Apply is logging DDLs that are skipped in the DBA_LOGSTDBY_EVENTS table ■ whether SQL Apply is logging DDLs that are applied in the DBA_LOGSTDBY_EVENTS table ■ whether SQL Apply is logging unsupported operations that are encountered in the DBA_LOGSTDBY_EVENTS table ■ whether or not real time apply is on ■ value of apply delay (in minutes) ■ coordinator state ■ coordinator uptime in seconds ■ time of the most recent start of SQL Apply ■ number of transactions mined and made available for apply ■ number of transactions applied ■ number of rolled back transactions mined ■ number of DDL txns mined ■ number of CTAS (Create Table as Select) txns mined ■ number of thread enable events encountered in the redo stream ■ number of thread disable events encountered in the redo stream ■ bytes of redo records mined ■ bytes paged out ■ seconds spent in pageout activity ■ bytes checkpointed ■ seconds spent in checkpointing activity ■ seconds SQL Apply is idle ■ number of times a complete standby redo logs are mined without having to mine the corresponding archived log ■ number of times SQL Apply had to switch from a standby redo log to the corresponding archived log ■ number of times SQL Apply mined redo from the archived logs ■ number of archived logs that arrived at the standby via gap fetch mechanism (gap fetched logs mined) ■ number of failed attempts to open a logfile ■ amount of time spent in waiting for the current gap to resolve if SQL Apply is running in real time mode (current logfile wait)[1] ■ time spent in waiting for gap to resolve if SQL Apply is running in real time mode (total logfile wait)[2]
VALUE	VARCHAR2(64)	Value of the statistic or state information

[1] In case SQL Apply is not running in real time mode, this may not reflect time spent in gap resolution, but simply the time spent waiting for the most recent archived log to show up at the logical standby.

[2] In case SQL Apply is not running in real time mode, this will include time that SQL Apply spent everytime it finished processing all archived logs registered with it, and waited for the next log to be archived.

V$LOGSTDBY_TRANSACTION

`V$LOGSTDBY_TRANSACTION` displays all transactions that are actively being processed by SQL Apply. The transaction identifiers shown in this view are those mined from the redo stream and correspond to transaction identifiers assigned at the primary database, and do not correspond to the transactions that are active at the logical standby database. For information regarding transactions active in the logical standby database, including those created as part of SQL Apply, query the `V$TRANSACTION` view at the logical standby database.

Column	Datatype	Description
PRIMARY_XIDUSN	NUMBER	Undo segment number of the transaction
PRIMARY_XIDSLT	NUMBER	Slot number of the transaction
PRIMARY_XIDSQN	NUMBER	Sequence number of the transaction
PRIMARY_XID	RAW(8)	Transaction ID
PRIMARY_START_SCN	NUMBER	Start system change number (SCN) base
PRIMARY_START_TIME	DATE	Start time
PRIMARY_PARENT_XIDUSN	NUMBER	Undo segment number of the parent transaction
PRIMARY_PARENT_XIDSLT	NUMBER	Slot number of the parent transaction
PRIMARY_PARENT_XIDSQN	NUMBER	Sequence number of the parent transaction
PRIMARY_PARENT_XID	RAW(8)	Transaction ID of the parent transaction (PDML)
TYPE	VARCHAR2(32)	Type: ▪ `PL/SQL` - Transaction was done as part of a supported PL/SQL procedure ▪ `Direct Path Load` - Transaction is a direct path load ▪ `CTAS` - Transaction contains at least one `CREATE TABLE ... AS SELECT` operation ▪ `DDL` - Transaction contains one or more DDL operations ▪ `PDML Child` - Transaction is a child transaction ▪ `DML` - Transaction contains only DML operations
MINING_STATUS	VARCHAR2(32)	Mining status: ▪ `ACTIVE` - Transaction is still being mined by LogMiner. At least part of this transaction is ready to be applied or has already been applied. ▪ `COMPLETE` - Transaction is complete and ready to be applied. LogMiner has finished mining.
APPLY_STATUS	VARCHAR2(6)	▪ `ACTIVE` - Transaction has been assigned to an apply server. It is in one of the following states: – The transaction is being actively applied – The transaction is being held by an apply server waiting for certain events to occur – The transaction is being held by an apply server waiting for subsequent parts of this transaction ▪ `NONE` - Transaction has not yet been assigned to an apply server

Column	Datatype	Description
SID	NUMBER	Session ID of the apply server's session; Null if APPLY_STATUS is NONE
SERIAL#	NUMBER	Serial number of the apply server's session; Null if APPLY_STATUS is NONE

V$MANAGED_STANDBY

V$MANAGED_STANDBY displays current status information for some Oracle Database processes related to physical standby databases in the Data Guard environment. This view does not persist after an instance shutdown.

Column	Datatype	Description
PROCESS	VARCHAR2(9)	Type of the process whose information is being reported: ■ RFS - Remote file server ■ MRP0 - Detached recovery server process ■ MR(fg) - Foreground recovery session ■ ARCH - Archiver process ■ FGRD ■ LGWR ■ RFS(FAL) ■ RFS(NEXP) ■ LNS - Network server process
PID	NUMBER	Operating system process identifier of the process
STATUS	VARCHAR2(12)	Current process status: ■ UNUSED - No active process ■ ALLOCATED - Process is active but not currently connected to a primary database ■ CONNECTED - Network connection established to a primary database ■ ATTACHED - Process is actively attached and communicating to a primary database ■ IDLE - Process is not performing any activities ■ ERROR - Process has failed ■ OPENING - Process is opening the archived redo log ■ CLOSING - Process has completed archival and is closing the archived redo log ■ WRITING - Process is actively writing redo data to the archived redo log ■ RECEIVING - Process is receiving network communication ■ ANNOUNCING - Process is announcing the existence of a potential dependent archived redo log ■ REGISTERING - Process is registering the existence of a completed dependent archived redo log ■ WAIT_FOR_LOG - Process is waiting for the archived redo log to be completed ■ WAIT_FOR_GAP - Process is waiting for the archive gap to be resolved ■ APPLYING_LOG - Process is actively applying the archived redo log to the standby database
CLIENT_PROCESS	VARCHAR2(8)	Identifies the corresponding primary database process: ■ Archival - Foreground (manual) archival process (SQL) ■ ARCH - Background ARC*n* process ■ LGWR - Background LGWR process
CLIENT_PID	VARCHAR2(40)	Operating system process identifier of the client process

Column	Datatype	Description
CLIENT_DBID	VARCHAR2(40)	Database identifier of the primary database
GROUP#	VARCHAR2(40)	Standby redo log group
RESETLOG_ID	NUMBER	Resetlogs identifier of the archived redo log
THREAD#	NUMBER	Archived redo log thread number
SEQUENCE#	NUMBER	Archived redo log sequence number
BLOCK#	NUMBER	Last processed archived redo log block number
BLOCKS	NUMBER	Size of the archived redo log in 512-byte blocks
DELAY_MINS	NUMBER	Archived redo log delay interval in minutes
KNOWN_AGENTS	NUMBER	Total number of standby database agents processing an archived redo log
ACTIVE_AGENTS	NUMBER	Number of standby database agents actively processing an archived redo log

V$MAP_COMP_LIST

V$MAP_COMP_LIST displays supplementary information for all element mapping structures.

Column	Datatype	Description
ELEM_IDX	NUMBER	Index corresponding to the element
NUM_COMP	NUMBER	Number of components (maximum is 5)
COMP1_NAME	VARCHAR2(256)	Name of the first component
COMP1_VAL	VARCHAR2(256)	Value of the first component
COMP2_NAME	VARCHAR2(256)	Name of the second component
COMP2_VAL	VARCHAR2(256)	Value of the second component
COMP3_NAME	VARCHAR2(256)	Name of the third component
COMP3_VAL	VARCHAR2(256)	Value of the third component
COMP4_NAME	VARCHAR2(256)	Name of the fourth component
COMP4_VAL	VARCHAR2(256)	Value of the fourth component
COMP5_NAME	VARCHAR2(256)	Name of the fifth component
COMP5_VAL	VARCHAR2(256)	Value of the fifth component

V$MAP_ELEMENT

V$MAP_ELEMENT displays a list of all element mapping structures in the SGA of the instance.

Column	Datatype	Description
ELEM_NAME	VARCHAR2(256)	Element name
ELEM_IDX	NUMBER	Index corresponding to the element
ELEM_CFGID	VARCHAR2(256)	Configuration ID (N/A if configuration ID is not supported)

Column	Datatype	Description
ELEM_TYPE	VARCHAR2(12)	Element type: ▪ MIRROR ▪ STRIPE ▪ RAID5 ▪ CONCATENATED ▪ PARTITION ▪ DISK ▪ NONE
ELEM_SIZE	NUMBER	Element Size in HKB
ELEM_NSUBELEM	NUMBER	Number of Subelements
ELEM_DESCR	VARCHAR2(256)	Element Description
STRIPE_SIZE	NUMBER	Stripe Size in HKB for RAID-5 and STRIPE elements, 0 for the remaining types
LIB_IDX	NUMBER	Index of the library which claims ownership of the element

V$MAP_EXT_ELEMENT

V$MAP_EXT_ELEMENT displays supplementary information for all element mapping structures.

Column	Datatype	Description
ELEM_IDX	NUMBER	Index corresponding to the element
NUM_ATTRB	NUMBER	Number of Attributes (maximum is 5)
ATTRB1_NAME	VARCHAR2(256)	Name of the first Attribute
ATTRB1_VAL	VARCHAR2(256)	Value of the first attribute
ATTRB2_NAME	VARCHAR2(256)	Name of the second attribute
ATTRB2_VAL	VARCHAR2(256)	Value of the second attribute
ATTRB3_NAME	VARCHAR2(256)	Name of the third attribute
ATTRB3_VAL	VARCHAR2(256)	Value of the third attribute
ATTRB4_NAME	VARCHAR2(256)	Name of the fourth attribute
ATTRB4_VAL	VARCHAR2(256)	Value of the fourth attribute
ATTRB5_NAME	VARCHAR2(256)	Name of the fifth attribute
ATTRB5_VAL	VARCHAR2(256)	Value of the fifth attribute

V$MAP_FILE

V$MAP_FILE displays a list of all file mapping structures in the shared memory of the instance.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	Index corresponding to the file
FILE_CFGID	VARCHAR2(256)	Configuration ID (N/A if configuration ID is not supported)
FILE_STATUS	VARCHAR2(7)	Status of the mapping information: ▪ VALID - File mapping information is latest ▪ INVALID - Mapping needs to be refreshed
FILE_NAME	VARCHAR2(256)	Absolute file name

Column	Datatype	Description
FILE_TYPE	VARCHAR2(11)	File type: ■ DATAFILE ■ SPFILE ■ TEMPFILE ■ CONTROLFILE ■ LOGFILE ■ ARCHIVEFILE
FILE_STRUCTURE	VARCHAR2(9)	File structure: ■ FILE ■ RAWVOLUME ■ RAWDEVICE ■ NONE
FILE_SIZE	NUMBER	File size in HKB (half KB)
FILE_NEXTS	NUMBER	Number of file extents in the file (not necessarily the same as the number of file extents mapped)
LIB_IDX	NUMBER	Index of mapping library claiming ownership of the file

V$MAP_FILE_EXTENT

V$MAP_FILE_EXTENT displays a list of all file extent mapping structures in the shared memory of the instance.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	File index (corresponds to FILE_MAP_IDX in V$MAP_FILE)
EXT_NUM	NUMBER	File extent number
EXT_ELEM_OFF	NUMBER	Element offset in HKB
EXT_SIZE	NUMBER	File extent size in HKB
EXT_FILE_OFF	NUMBER	File Offset in HKB
EXT_TYPE	VARCHAR2(6)	File Extent Type: ■ DATA ■ PARITY ■ NONE
ELEM_IDX	NUMBER	Index in V$MAP_ELEMENT corresponding to the element where the file extent resides

V$MAP_FILE_IO_STACK

V$MAP_FILE_IO_STACK displays the hierarchical arrangement of storage containers for files. Each row in the view represents a level in the hierarchy.

Column	Datatype	Description
FILE_MAP_IDX	NUMBER	File index (corresponds to FILE_MAP_IDX in V$MAP_FILE)
DEPTH	NUMBER	Element depth within the I/O stack
ELEM_IDX	NUMBER	Index corresponding to the element
CU_SIZE	NUMBER	Contiguous set of logical blocks of the file (in HKB units) that is resident contiguously on the element
STRIDE	NUMBER	Number of HKB between contiguous units (CU) in the file that are contiguous on this element. Used in RAID5 and striped files.

Column	Datatype	Description
NUM_CU	NUMBER	Number of contiguous units that are adjacent to each other on this element that are separated by STRIDE HKB in the file. In RAID5, the number of contiguous units also include the parity stripes.
ELEM_OFFSET	NUMBER	Element offset in HKB units
FILE_OFFSET	NUMBER	Offset in HKB units from the start of the file to the first byte of the contiguous units
DATA_TYPE	VARCHAR2(15)	Datatype: ■ DATA ■ PARITY ■ DATA AND PARITY
PARITY_POS	NUMBER	Position of the parity. Only for RAID5. This column is needed to distinguish the parity from the data part.
PARITY_PERIOD	NUMBER	Parity period. Only for RAID5.
ID	NUMBER	Unique identifier
PARENT_ID	NUMBER	Parent identifier

V$MAP_LIBRARY

V$MAP_LIBRARY displays a list of all mapping libraries dynamically loaded by the external process.

Column	Datatype	Description
LIB_IDX	NUMBER	Index corresponding to the library
LIB_NAME	VARCHAR2(256)	Absolute library name
VENDOR_NAME	VARCHAR2(64)	Name of the vendor implementing the library
PROTOCOL_NUM	NUMBER	Mapping protocol that the library supports
VERSION_NUM	VARCHAR2(32)	Version number
PATH_NAME	VARCHAR2(4000)	Path name
MAP_FILE	VARCHAR2(1)	Indicates whether the library supports mapping files (Y) or not (N)
FILE_CFGID	VARCHAR2(13)	Type of configuration ID supported for files: ■ NONE - Not supported ■ PERSISTENT ■ NONPERSISTENT
MAP_ELEM	VARCHAR2(1)	Indicates whether the library supports mapping elements (Y) or not (N)
ELEM_CFGID	VARCHAR2(13)	Type of configuration id supported for elements: ■ NONE - Not supported ■ PERSISTENT ■ NONPERSISTENT
MAP_SYNC	VARCHAR2(1)	Indicates whether the library needs to be explicitly synced so that future mappings reflect the most recent changes (Y) or not (N). Note that configuration IDs cannot be supported if the library needs to be explicitly synced.

V$MAP_SUBELEMENT

V$MAP_SUBELEMENT displays a list of all subelement mapping structures in the shared memory of the instance.

Column	Datatype	Description
CHILD_IDX	NUMBER	Index in V$MAP_ELEMENT corresponding to the child element
PARENT_IDX	NUMBER	Index in V$MAP_ELEMENT corresponding to the parent element
SUB_NUM	NUMBER	Subelement number
SUB_SIZE	NUMBER	Subelement size in HKB
ELEM_OFFSET	NUMBER	Offset in HKB on the child element
SUB_FLAGS	NUMBER	Subelement flags (currently unused)

V$MEMORY_CURRENT_RESIZE_OPS

V$MEMORY_CURRENT_RESIZE_OPS displays information about memory resize operations (both automatic and manual) which are currently in progress. An operation can be a grow or a shrink of a dynamic memory component. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Desired value of the parameter after the resize
CURRENT_SIZE	NUMBER	Current value of the parameter
START_TIME	DATE	Start time of the operation
LAST_UPDATE_TIME	DATE	Last time progress was made for the operation

V$MEMORY_DYNAMIC_COMPONENTS

V$MEMORY_DYNAMIC_COMPONENTS displays information about the dynamic SGA components. This view summarizes information based on all completed SGA resize operations since instance startup. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
CURRENT_SIZE	NUMBER	Current size of the component
MIN_SIZE	NUMBER	Minimum size of the component since instance startup
MAX_SIZE	NUMBER	Maximum size of the component since instance startup
USER_SPECIFIED_SIZE	NUMBER	Value of the user parameter for the component
OPER_COUNT	NUMBER	Number of operations since instance startup

Column	Datatype	Description
LAST_OPER_TYPE	VARCHAR2(13)	Last completed operation for the component: - STATIC - INITIALIZING - DISABLED - GROW - SHRINK - SHRINK_CANCEL
LAST_OPER_MODE	VARCHAR2(9)	Mode of the last completed operation: - MANUAL - DEFERRED - IMMEDIATE
LAST_OPER_TIME	DATE	Start time of the last completed operation
GRANULE_SIZE	NUMBER	Granularity of the GROW or SHRINK operation

V$MEMORY_RESIZE_OPS

V$MEMORY_RESIZE_OPS displays information about the last 800 completed memory resize operations (both automatic and manual). This does not include in-progress operations. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: - STATIC - INITIALIZING - DISABLED - GROW - SHRINK - SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: - MANUAL - DEFERRED - IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Requested value of the parameter after the resize
FINAL_SIZE	NUMBER	Real value of the parameter after the resize
STATUS	VARCHAR2(9)	Completion status of the operation: - INACTIVE - PENDING - COMPLETE - CANCELLED - ERROR
START_TIME	DATE	Start time of the operation
END_TIME	DATE	End time of the operation

V$MEMORY_TARGET_ADVICE

V$MEMORY_TARGET_ADVICE provides information about how the MEMORY_TARGET parameter should be sized based on current sizing and satisfaction metrics.

Column	Datatype	Description
MEMORY_SIZE	NUMBER	If the MEMORY_SIZE_FACTOR column has a value of 1, then this column shows the current size of memory, as set by the MEMORY_TARGET initialization parameter. If the value of the MEMORY_SIZE_FACTOR column is less than or greater than 1, then this column shows a proposed memory size.
MEMORY_SIZE_FACTOR	NUMBER	A multiplier for the current memory size. Possible values are 0.25, 0.5, 0.75, 1, 1.5, 1.75, and 2. This multiplier times the current memory size equals the value of the MEMORY_SIZE column.
ESTD_DB_TIME	NUMBER	For current memory size (MEMORY_SIZE_FACTOR = 1), the amount of database time required to complete the current workload. For a proposed memory size, the estimated amount of database time that would be required if the MEMORY_TARGET parameter were changed to the proposed size.
ESTD_DB_TIME_FACTOR	NUMBER	For a proposed memory size, ratio of estimated database time to current database time
VERSION	NUMBER	Version number of this recommendation (this snapshot of the V$MEMORY_TARGET_ADVICE view)

Table 7–2 shows how the information provided in V$MEMORY_TARGET_ADVICE could be used to improve performance. The data indicates that if current memory size is 380M, and you were to increase it to 760M (2x), the current workload would take 80525 units of DBtime as opposed to 115475 units of DBtime, which is a significant improvement in performance.

Table 7–2 Example of Using V$MEMORY_TARGET_ADVICE

MEMORY_SIZE	MEMORY_SIZE_ FACTOR	ESTD_DB_TIME	ESTD_DB_TIME_ FACTOR	VERSION
380	1	115475	1	3
95	.25	200500	1.7	3
190	.5	125600	1.1	3
760	2	80525	0.7	3

V$METRIC

V$METRIC displays the most recent statistic values for the complete set of metrics captured by the AWR infrastructure.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time for the metric value
END_TIME	DATE	End time for the metric value
INTSIZE_CSEC	NUMBER	Size of the time period
GROUP_ID	NUMBER	Metric Group ID. Refer to the V$METRICNAME view for the name of the group.
ENTITY_ID	NUMBER	Entity ID for the metric in question. The value of the Entity ID depends upon the metric group. (See the following table for an explanation of possible values.)

Column	Datatype	Description
ENTITY_SEQUENCE	NUMBER	Entity Sequence number for the metric in question. The value of the Entity Sequence depends upon the metric group. (See the following table for an explanation of possible values.)
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric Name. This is the statistic that is captured for the entity.
VALUE	NUMBER	Value of the statistic between BEGIN_TIME and END_TIME
METRIC_UNIT	VARCHAR2(64)	Unit for the VALUE

The following table describes what the ENTITY_ID and ENTITY_SEQUENCE are for each metric group:

GID	Group NAME	Entity ID	Entity Sequence
0	Event Metrics	Event#	N/A
1	Event Class Metrics	Wait Class ID	N/A
2	System Metrics Long Duration	N/A	N/A
3	System Metrics Short Duration	N/A	N/A
4	Session Metrics Long Duration	Session ID	Serial#
5	Session Metrics Short Duration	Session ID	Serial#
6	Service Metrics	N/A	Service Hash
7	File Metrics Long Duration	File#	Creation Change#
9	Tablespace Metrics Long Duration	Tablespace#	N/A
10	Service Metrics (Short)	N/A	Service Hash

V$METRICGROUP

V$METRICGROUP displays information about the metric group for each of the four major Streams components: capture, propagation, apply, and queue.

Column	Datatype	Description
GROUP_ID	NUMBER	Internal ID associated with each group
NAME	VARCHAR2(64)	External name of the group
INTERVAL_SIZE	NUMBER	How often to collect statistics
MAX_INTERVAL	NUMBER	Total number of intervals over which statistics should be collected

V$METRIC_HISTORY

V$METRIC_HISTORY displays all the available statistic values for the complete set of metrics captured by the AWR infrastructure.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time for the metric value
END_TIME	DATE	End time for the metric value
INTSIZE_CSEC	NUMBER	Size of the time period
GROUP_ID	NUMBER	Metric Group ID. Refer to the V$METRICNAME view for the name of the group.

Column	Datatype	Description
ENTITY_ID	NUMBER	Entity ID for the metric in question. The value of the Entity ID depends upon the metric group. (See the table in the description of V$METRIC for an explanation of possible values.)
ENTITY_SEQUENCE	NUMBER	Entity Sequence number for the metric in question. The value of the Entity Sequence depends upon the metric group. (See the table in the description of V$METRIC for an explanation of possible values.)
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric Name. This is the statistic that is captured for the entity.
VALUE	NUMBER	Value of the statistic between BEGIN_TIME and END_TIME
METRIC_UNIT	VARCHAR2(64)	Unit for the VALUE

V$METRICNAME

V$METRICNAME displays the mapping of the name of metrics to their metric ID.

Column	Datatype	Description
GROUP_ID	NUMBER	Metric group ID
GROUP_NAME	VARCHAR2(64)	Metric group name
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric name
METRIC_UNIT	VARCHAR2(64)	Unit of measurement

V$MTTR_TARGET_ADVICE

V$MTTR_TARGET_ADVICE displays rows that predict the number of physical I/Os for the MTTR corresponding to each row. The rows also compute a physical I/O factor, which is the ratio of the number of estimated I/Os to the number of I/Os actually performed by the current MTTR setting during the measurement interval.

The content of the view is empty if MTTR advisory has not been turned on since database startup. Otherwise, it returns the advisory information collected. If advisory is currently off, then this information comes from the last time MTTR advisory was on. FAST_START_MTTR_TARGET must be set to a nonzero value if the STATISTICS_LEVEL parameter is dynamically modified to turn MTTR advisory on.

If the FAST_START_MTTR_TARGET parameter is changed while MTTR advisory is on, then MTTR advisory is temporarily turned off until the new FAST_START_MTTR_TARGET setting takes effect. During this transition period, the contents of V$MTTR_TARGET_ADVICE reflect the simulation result for the old MTTR setting.

Column	Datatype	Description
MTTR_TARGET_FOR_ESTIMATE	NUMBER	MTTR setting being simulated. Equal to the current MTTR setting if this is the first row of the view.
ADVICE_STATUS	VARCHAR2(5)	Current status of MTTR simulation: ■ ON ■ READY ■ OFF
DIRTY_LIMIT	NUMBER	Dirty buffer limit derived from the MTTR being simulated
ESTD_CACHE_WRITES	NUMBER	Estimated number of cache physical writes under this MTTR

Column	Datatype	Description
ESTD_CACHE_WRITE_FACTOR	NUMBER	Estimated cache physical write ratio under this MTTR. It is the ratio of the estimated number of cache writes to the number of cache writes under the current MTTR setting.
ESTD_TOTAL_WRITES	NUMBER	Estimated total number of physical writes under this MTTR
ESTD_TOTAL_WRITE_FACTOR	NUMBER	Estimated total physical write ratio under this MTTR. It is the ratio of the estimated total number of physical writes to the total number of physical writes under the current MTTR setting.
ESTD_TOTAL_IOS	NUMBER	Estimated total number of I/Os under this MTTR
ESTD_TOTAL_IO_FACTOR	NUMBER	Estimated total I/O ratio under this MTTR. It is the ratio of the estimated total number of I/Os to the total number of I/Os under the current MTTR setting.

V$MUTEX_SLEEP

V$MUTEX_SLEEP shows the wait time, and the number of sleeps for each combination of mutex type and location.

Column	Datatype	Description
MUTEX_TYPE	VARCHAR2(32)	Type of action/object the mutex protects
LOCATION	VARCHAR2(40)	The code location where the waiter slept for the mutex
SLEEPS	NUMBER	Number of sleeps for this MUTEX_TYPE and LOCATION
WAIT_TIME	NUMBER	Wait time in microseconds

V$MUTEX_SLEEP_HISTORY

V$MUTEX_SLEEP_HISTORY displays time-series data. Each row in this view is for a specific time, mutex type, location, requesting session and blocking session combination. That is, it shows data related to a specific session (requesting session) that slept while requesting a specific mutex type and location, because it was being held by a specific blocking session. The data in this view is contained within a circular buffer, with the most recent sleeps shown.

Column	Datatype	Description
MUTEX_IDENTIFIER	NUMBER	Mutex ID
SLEEP_TIMESTAMP	TIMESTAMP(6)	The last date/time this MUTEX_TYPE and LOCATION was slept for by the REQUESTING_SESSION, while being held by the BLOCKING_SESSION.
MUTEX_TYPE	VARCHAR2(32)	Type of action/object the mutex protects
GETS	NUMBER	The number of times the mutex/location was requested by the requesting session while being held by the blocking session. GETS is only incremented once per request, irrespective of the number of sleeps required to obtain the mutex.
SLEEPS	NUMBER	The number of times the requestor had to sleep before obtaining the mutex
REQUESTING_SESSION	NUMBER	The SID of a session requesting the mutex
BLOCKING_SESSION	NUMBER	The SID of a session holding the mutex
LOCATION	VARCHAR2(40)	The code location where the waiter slept for the mutex
MUTEX_VALUE	RAW(4)	If the mutex is held in exclusive (X) mode, this column shows the SID of the blocking session, else it shows the number of sessions referencing the mutex in S mode.
P1	NUMBER	Internal use only
P1RAW	RAW(4)	Internal use only

Column	Datatype	Description
P2	NUMBER	Internal use only
P3	NUMBER	Internal use only
P4	NUMBER	Internal use only
P5	VARCHAR2(64)	Internal use only

V$MVREFRESH

V$MVREFRESH displays information about the materialized views currently being refreshed.

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number, which is used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends with, and another session begins with, the same session ID.
CURRMVOWNER	VARCHAR2(31)	Owner of the materialized view currently being refreshed. The materialized view resides in this user's schema.
CURRMVNAME	VARCHAR2(31)	Name of the materialized view currently being refreshed

V$MYSTAT

V$MYSTAT contains statistics on the current session.

Column	Datatype	Description
SID	NUMBER	ID of the current session
STATISTIC#	NUMBER	Number of the statistic
VALUE	NUMBER	Value of the statistic

V$NFS_CLIENTS

V$NFS_CLIENTS displays information about NFS clients currently connected to the XML DB NFS Server.

Column	Datatype	Description
CLIENTID	NUMBER	A number identifying the client
PRINCIPAL	VARCHAR2(2000)	User string denoting the principal that set the client ID (SetClientId)
CLIENTOPAQUEIDENTIFIER	VARCHAR2(1000)	Opaque string presented as identification by the client to the NFS server
VERIFIER	RAW(8)	Verifier presented by the client
LEASEEXPIRY	NUMBER	Number of seconds in which the lease expires for the client
CLIENTADDR	VARCHAR2(2000)	Address of the client
CONFIRMED	VARCHAR2(5)	TRUE if the client is confirmed; otherwise FALSE

V$NFS_LOCKS

V$NFS_LOCKS displays information about byte range locks held on different files by NFS clients.

Column	Datatype	Description
OPENSTATEID	RAW(16)	Open state ID of the open owner
OPENSEQUENCEID	NUMBER	Open Sequence ID of the open owner
LOCKSTATEID	RAW(16)	Lock state ID of the lock owner
LOCKSEQUENCEID	NUMBER	Lock sequence ID of the lock owner
LOCKOWNER	VARCHAR2(2000)	Opaque string presented as identification by the lock owner to the NFS server
OFFSET	NUMBER	Byte Offset from which lock starts
LENGTH	NUMBER	Length of the lock
LOCKTYPE	VARCHAR2(20)	Type of the lock

V$NFS_OPEN_FILES

V$NFS_OPEN_FILES displays information about all the files currently opened by clients at the NFS server.

Column	Datatype	Description
CLIENTID	NUMBER	Number identifying the client
OPENOWNEROPAQUE	VARCHAR2(2000)	All the files currently opened by clients at the NFS server
OPENSTATEID	RAW(16)	Open state ID of the open owner
FILEHANDLE	RAW(32)	FileHandle of the file that has been opened
OPENSEQUENCEID	NUMBER	Open sequence ID of open owner
OPENREAD	VARCHAR2(5)	TRUE if the file is open for READ operations; otherwise FALSE
OPENWRITE	VARCHAR2(5)	TRUE if the file is open for WRITE operations; otherwise FALSE
SHAREACCESS	VARCHAR2(15)	Sharing mode of the file (SharedReadWrite, SharedRead, SharedWrite)
SHAREDENY	VARCHAR2(13)	Deny mode of the file (DenyReadWrite, DenyRead, DenyWrite)
CONFIRMED	VARCHAR2(5)	TRUE if open is confirmed; otherwise FALSE

V$NLS_PARAMETERS

V$NLS_PARAMETERS contains current values of NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Parameter names are as follows: NLS_CALENDAR, NLS_CHARACTERSET, NLS_COMP, NLS_CURRENCY, NLS_DATE_FORMAT, NLS_DATE_LANGUAGE, NLS_DUAL_CURRENCY, NLS_ISO_CURRENCY, NLS_LANGUAGE, NLS_LENGTH_SEMANTICS, NLS_NCHAR_CHARACTERSET, NLS_NCHAR_CONV_EXCP, NLS_NUMERIC_CHARACTERS, NLS_SORT, NLS_TERRITORY, NLS_TIMESTAMP_FORMAT, NLS_TIMESTAMP_TZ_FORMAT Two additional parameters, NLS_TIME_FORMAT and NLS_TIME_TZ_FORMAT, are currrently used for internal purposes only.
VALUE	VARCHAR2(64)	NLS parameter value

V$NLS_VALID_VALUES

V$NLS_VALID_VALUES lists all valid values for NLS parameters.

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Parameter name (LANGUAGE | SORT | TERRITORY | CHARACTERSET)

Column	Datatype	Description
VALUE	VARCHAR2(64)	NLS parameter value
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)

V$OBJECT_DEPENDENCY

V$OBJECT_DEPENDENCY displays the objects depended on by a package, procedure, or cursor that is currently loaded in the shared pool. For example, together with V$SESSION and V$SQL, this view can be used to determine which tables are used in the SQL statement that a user is currently executing.

See Also: "V$SESSION" on page 8-8 and "V$SQL" on page 8-26

Column	Datatype	Description
FROM_ADDRESS	RAW(4 | 8)	Address of a procedure, package, or cursor that is currently loaded in the shared pool
FROM_HASH	NUMBER	Hash value of a procedure, package, or cursor that is currently loaded in the shared pool
TO_OWNER	VARCHAR2(64)	Owner of the object that is depended on
TO_NAME	VARCHAR2(1000)	Name of the object that is depended on
TO_ADDRESS	RAW(4 | 8)	Address of the object that is depended on. These can be used to look up more information on the object in V$DB_OBJECT_CACHE.
TO_HASH	NUMBER	Hash value of the object that is depended on. These can be used to look up more information on the object in V$DB_OBJECT_CACHE.
TO_TYPE	NUMBER	Type of the object that is depended on

V$OBJECT_PRIVILEGE

V$OBJECT_PRIVILEGE displays information about privileges associated with an object.

Column	Datatype	Description
OBJECT_TYPE_NAME	VARCHAR2(64)	Name of the object type
OBJECT_TYPE_ID	NUMBER	ID of the object type
PRIVILEGE_ID	NUMBER	ID of the privilege
PRIVILEGE_NAME	VARCHAR2(64)	Name of the privilege

V$OBJECT_USAGE

V$OBJECT_USAGE displays statistics about index usage gathered from the database. You can use this view to monitor index usage. All indexes that have been used at least once can be monitored and displayed in this view.

Column	Datatype	Description
INDEX_NAME	VARCHAR2(30)	Index name in sys.obj$.name
TABLE_NAME	VARCHAR2(30)	Table name in sys.obj$.name
MONITORING	VARCHAR2(3)	YES | NO
USED	VARCHAR2(3)	YES | NO
START_MONITORING	VARCHAR2(19)	Start monitoring time in sys.object_stats.start_monitoring

Column	Datatype	Description
END_MONITORING	VARCHAR2(19)	End monitoring time in sys.object_stats.end_monitoring

V$OBSOLETE_BACKUP_FILES

V$OBSOLETE_BACKUP_FILES displays all obsolete backups, copies, and archived logs according to the current retention policy. This view requires that the database is set using the DBMS_RCVMAN.SETDATABASE procedure.

Column	Datatype	Description
PKEY	NUMBER	Primary key for the backup
BACKUP_TYPE	VARCHAR2(32)	Type of the backup: ■ BACKUP SET ■ COPY ■ PROXY COPY
FILE_TYPE	VARCHAR2(32)	Type of the file: ■ DATAFILE ■ CONTROLFILE ■ SPFILE ■ REDO LOG ■ PIECE
KEEP	VARCHAR2(3)	Indicates whether the backup has a retention policy different from the value for CONFIGURE RETENTION POLICY (YES) or not (NO)
KEEP_UNTIL	DATE	If the KEEP UNTIL TIME clause of the BACKUP command was specified, then this column shows the date after which the backup becomes obsolete. If the column is null and KEEP_OPTIONS is not null, the backup never becomes obsolete.
KEEP_OPTIONS	VARCHAR2(13)	KEEP options for the backup: ■ LOGS - RMAN keeps the logs needed to recover the backup ■ NOLOGS - RMAN does not keep the logs needed to recover the backup If this column is null, then the backup has no KEEP options and will be made obsolete based on the retention policy.
STATUS	VARCHAR2(16)	Status of the backup: ■ AVAILABLE ■ UNAVAILABLE ■ EXPIRED ■ OTHER
FNAME	VARCHAR2(1024)	Name of the file
TAG	VARCHAR2(32)	Tag of the piece, copy, or proxy copy
MEDIA	VARCHAR2(80)	Media ID of the piece or proxy copy
RECID	NUMBER	Recid of the record in the controlfile
STAMP	NUMBER	Stamp of the record in the controlfile
DEVICE_TYPE	VARCHAR2(255)	Type of media device that stores the backup
BLOCK_SIZE	NUMBER	Block size for the backup (in bytes)
COMPLETION_TIME	DATE	Time when the backup completed
BS_KEY	NUMBER	Primary key of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_COUNT	NUMBER	Count of the backup set from the controlfile record (valid only when BACKUP_TYPE is BACKUP SET)

Column	Datatype	Description
BS_STAMP	NUMBER	Stamp of the backup set from the controlfile record (valid only when BACKUP_TYPE is BACKUP SET)
BS_TYPE	VARCHAR2(32)	Type of the backup set (valid only when BACKUP_TYPE is BACKUP SET): ■ DATAFILE ■ ARCHIVED LOG
BS_INCR_TYPE	VARCHAR2(32)	Incremental level of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_PIECES	NUMBER	Number of backup pieces in the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BS_COMPLETION_TIME	DATE	Completion time of the backup set (valid only when BACKUP_TYPE is BACKUP SET)
BP_PIECE#	NUMBER	Number of the backup piece (valid only when FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
BP_COPY#	NUMBER	Copy number of the backup piece (valid only when FILE_TYPE is PIECE and BACKUP_TYPE is BACKUP SET)
DF_FILE#	NUMBER	Absolute file number of the datafile (valid only when FILE_TYPE is DATAFILE)
DF_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the control file or datafile was created (valid only when FILE_TYPE is DATAFILE)
DF_CREATION_CHANGE#	NUMBER	Creation SCN of the control file or datafile (valid only when FILE_TYPE is DATAFILE)
DF_CHECKPOINT_CHANGE#	NUMBER	System change number (SCN) of the most recent control file or datafile checkpoint (valid only when FILE_TYPE is DATAFILE)
DF_CKP_MOD_TIME	DATE	Modification time in case of SPFILE, otherwise time when the control file or datafile was checkpointed (valid only when FILE_TYPE is DATAFILE)
RL_THREAD#	NUMBER	Number of the redo thread (valid only when FILE_TYPE is REDO LOG)
RL_SEQUENCE#	NUMBER	Log sequence number (valid only when FILE_TYPE is REDO LOG)
RL_RESETLOGS_CHANGE#	NUMBER	System change number (SCN) of the most recent RESETLOGS when the record was created (valid only when FILE_TYPE is REDO LOG)
RL_FIRST_CHANGE#	NUMBER	First SCN of the redo log (valid only when FILE_TYPE is REDO LOG)
RL_FIRST_TIME	DATE	Time when Oracle switched into the redo log (valid only when FILE_TYPE is REDO LOG)
RL_NEXT_CHANGE#	NUMBER	First SCN of the next redo log in the thread (valid only when FILE_TYPE is REDO LOG)
RL_NEXT_TIME	DATE	First timestamp of the next redo log in the thread (valid only when FILE_TYPE is REDO LOG)

V$OBSOLETE_PARAMETER

V$OBSOLETE_PARAMETER displays information about obsolete initialization parameters. If any row of the view contains TRUE in the ISSPECIFIED column, then you should examine why.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the parameter
ISSPECIFIED	VARCHAR2(5)	Indicates whether the parameter was specified in the parameter file (TRUE) or not (FALSE)

V$OFFLINE_RANGE

V$OFFLINE_RANGE displays datafile offline information from the control file. Note that the last offline range of each datafile is kept in the DATAFILE record.

An offline range is created for a datafile when its tablespace is first altered to be OFFLINE NORMAL or READ ONLY, and then subsequently altered to be ONLINE or read/write. Note that no offline range is created if the datafile itself is altered to be OFFLINE or if the tablespace is altered to be OFFLINE IMMEDIATE.

See Also: "V$DATAFILE" on page 6-64

Column	Datatype	Description
RECID	NUMBER	Record ID
STAMP	NUMBER	Record stamp
FILE#	NUMBER	Datafile number
OFFLINE_CHANGE#	NUMBER	SCN at which offlined
ONLINE_CHANGE#	NUMBER	SCN at which onlined
ONLINE_TIME	DATE	Time of offline SCN
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the record
RESETLOGS_TIME	DATE	Resetlogs timestamp of the record

V$OPEN_CURSOR

V$OPEN_CURSOR lists cursors that each user session currently has opened and parsed.

Column	Datatype	Description
SADDR	RAW(4 | 8)	Session address
SID	NUMBER	Session identifier
USER_NAME	VARCHAR2(30)	User that is logged in to the session
ADDRESS	RAW(4 | 8)	Used with HASH_VALUE to uniquely identify the SQL statement being executed in the session
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify the SQL statement being executed in the session
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement being executed in the session
SQL_TEXT	VARCHAR2(60)	First 60 characters of the SQL statement that is parsed into the open cursor
LAST_SQL_ACTIVE_TIME	DATE	Time when this cursor was last executed
SQL_EXEC_ID	NUMBER	If the open cursor is executing, then the SQL execution identifier for that execution (see V$SQL_MONITOR)

V$OPTION

V$OPTION lists database options and features. Typically, although not always, options must be separately purchased and installed, whereas features come with the product and are enabled based on the product that is running (Standard Edition, Enterprise Edition, or Personal Oracle).

Column	Datatype	Description
PARAMETER	VARCHAR2(64)	Name of the option (or feature)
VALUE	VARCHAR2(64)	Indicates whether the option (or feature) is installed (TRUE) or not (FALSE)

See Also: "DBA_REGISTRY" on page 5-18 for a list of all installed options for the database

V$OSSTAT

V$OSSTAT displays system utilization statistics from the operating system. One row is returned for each system statistic.

Column	Datatype	Description
STAT_NAME	VARCHAR2(64)	Name of the statistic (see Table 7–3)
VALUE	NUMBER	Instantaneous statistic value
OSSTAT_ID	NUMBER	Statistic ID
COMMENTS	VARCHAR2(64)	Any additional OS-specific clarifications for the statistic
CUMULATIVE	VARCHAR2(3)	Indicates whether the statistic is cumulative (that is, accumulates over time) (YES) or not (NO)

Table 7–3 V$OSSTAT Statistics

Statistic Name	Description
NUM_CPUS	Number of CPUs or processors available
IDLE_TIME	Number of hundredths of a second that a processor has been idle, totalled over all processors
BUSY_TIME	Number of hundredths of a second that a processor has been busy executing user or kernel code, totalled over all processors
USER_TIME	Number of hundredths of a second that a processor has been busy executing user code, totalled over all processors
SYS_TIME	Number of hundredths of a second that a processor has been busy executing kernel code, totalled over all processors
IOWAIT_TIME	Number of hundredths of a second that a processor has been waiting for I/O to complete, totalled over all processors
NICE_TIME	Number of hundredths of a second that a processor has been busy executing low-priority user code, totalled over all processors
AVG_IDLE_TIME	Number of hundredths of a second that a processor has been idle, averaged over all processors
AVG_BUSY_TIME	Number of hundredths of a second that a processor has been busy executing user or kernel code, averaged over all processors
AVG_USER_TIME	Number of hundredths of a second that a processor has been busy executing user code, averaged over all processors
AVG_SYS_TIME	Number of hundredths of a second that a processor has been busy executing kernel code, averaged over all processors
AVG_IOWAIT_TIME	Number of hundredths of a second that a processor has been waiting for I/O to complete, averaged over all processors
AVG_NICE_TIME	Number of hundredths of a second that a processor has been busy executing low-priority user code, averaged over all processors
OS_CPU_WAIT_TIME	Total number of hundredths of a second that processes have been in a ready state, waiting to be selected by the operating system scheduler to run
RSRC_MGR_CPU_WAIT_TIME	Total number of hundredths of a second that Oracle processes have been in a ready state, waiting for CPU to be available for their consumer group in the currently active resource plan
VM_PAGE_IN_BYTES	Total number of bytes of data that have been paged in due to virtual memory paging
VM_PAGE_OUT_BYTES	Total number of bytes of data that have been paged out due to virtual memory paging
PHYSICAL_MEMORY_BYTES	Total number of bytes of physical memory

Table 7–3 (Cont.) V$OSSTAT Statistics

Statistic Name	Description
LOAD	Current number of processes that are either running or in the ready state, waiting to be selected by the operating-system scheduler to run. On many platforms, this statistic reflects the average load over the past minute.
NUM_CPU_CORES	Number of CPU cores available (includes subcores of multicore CPUs as well as single-core CPUs)
NUM_CPU_SOCKETS	Number of CPU sockets available (represents an absolute count of CPU chips on the system, regardless of multithreading or multi-core architectures)
TCP_SEND_SIZE_MIN	Minimum size of the TCP send buffer
TCP_SEND_SIZE_DEFAULT	Default size of the TCP send buffer
TCP_SEND_SIZE_MAX	Maximum size of the TCP send buffer
TCP_RECEIVE_SIZE_MIN	Minimum size of the TCP receive buffer
TCP_RECEIVE_SIZE_DEFAULT	Default size of the TCP receive buffer
TCP_RECEIVE_SIZE_MAX	Maximum size of the TCP receive buffer
GLOBAL_SEND_SIZE_MAX	Maximum size of the global send buffer
GLOBAL_RECEIVE_SIZE_MAX	Maximum size of the global receive buffer

> **Note:** The availability of all statistics except for `NUM_CPUS` and `RSRC_MGR_CPU_WAIT_TIME` is subject to the operating system platform on which the Oracle Database is running.

V$PARALLEL_DEGREE_LIMIT_MTH

`V$PARALLEL_DEGREE_LIMIT_MTH` displays all available parallel degree limit resource allocation methods.

Column	Datatype	Description
`NAME`	`VARCHAR2(40)`	Name of the parallel degree limit resource allocation method

V$PARAMETER

`V$PARAMETER` displays information about the initialization parameters that are currently in effect for the session. A new session inherits parameter values from the instance-wide values displayed by the `V$SYSTEM_PARAMETER` view.

Column	Datatype	Description
`NUM`	`NUMBER`	Parameter number
`NAME`	`VARCHAR2(80)`	Name of the parameter
`TYPE`	`NUMBER`	Parameter type: ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
`VALUE`	`VARCHAR2(4000)`	Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value

Column	Datatype	Description
DISPLAY_VALUE	VARCHAR2(4000)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(9)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: ■ IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEm, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
ISBASIC	VARCHAR2(5)	Indicates whether the parameter is a basic parameter (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update
HASH	NUMBER	Hash value for the parameter name

V$PARAMETER_VALID_VALUES

V$PARAMETER_VALID_VALUES displays a list of valid values for list parameters.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(64)	Parameter name
ORDINAL	NUMBER	Ordinal number in the list (1-based)
VALUE	VARCHAR2(255)	Parameter value at ordinal
ISDEFAULT	VARCHAR2(64)	Indicates whether the given ordinal value is the default value for the parameter

V$PARAMETER2

V$PARAMETER2 displays information about the initialization parameters that are currently in effect for the session, with each list parameter value appearing as a row in the view. A new session inherits parameter values from the instance-wide values displayed in the V$SYSTEM_PARAMETER2 view.

Presenting the list parameter values in this format enables you to quickly determine the values for a list parameter. For example, if a parameter value is a, b, then the V$PARAMETER view does not tell you if the parameter has two values (both a and b) or one value (a, b). V$PARAMETER2 makes the distinction between the list parameter values clear.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(4000)	Parameter value for the session (if modified within the session); otherwise, the instance-wide parameter value
DISPLAY_VALUE	VARCHAR2(4000)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(6)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: ■ IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEm, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(10)	Indicates whether the parameter has been modified after instance startup: ■ MODIFIED - Parameter has been modified with ALTER SESSION ■ SYSTEM_MOD - Parameter has been modified with ALTER SYSTEM (which causes all the currently logged in sessions' values to be modified) ■ FALSE - Parameter has not been modified after instance startup

Column	Datatype	Description
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
ISBASIC	VARCHAR2(5)	Indicates whether the parameter is a basic parameter (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
ORDINAL	NUMBER	Position (ordinal number) of the parameter value. Useful only for parameters whose values are lists of strings.
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update

V$PERSISTENT_PUBLISHERS

V$PERSISTENT_PUBLISHERS displays information about all active publishers of the persistent queues in the database. There is one row per instance per queue per publisher. The rows are deleted when the database (or instance in an Oracle RAC environment) restarts.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
PUBLISHER_NAME	VARCHAR2(30)	Name of the agent enqueuing the message
PUBLISHER_ADDRESS	VARCHAR2(1024)	Address of the publisher agent
PROTOCOL	NUMBER	Protocol used by the publisher's address
ENQUEUED_MSGS	NUMBER	Number of messages that have been enqueued
ELAPSED_ENQUEUE_TIME	NUMBER	Total time spent doing enqueue(in hundredths of a second)
LAST_ENQUEUE_TIME	TIMESTAMP(6)	Last enqueue message timestamp

V$PERSISTENT_QUEUES

V$PERSISTENT_QUEUES displays information about all active persistent queues in the database since the queues' first activity time. There is one row per queue. The rows are deleted when the database (or instance in an Oracle RAC environment) restarts.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
FIRST_ACTIVITY_TIME	TIMESTAMP(6)	First queue activity time since database startup
ENQUEUED_MSGS	NUMBER	Number of messages enqueued
DEQUEUED_MSGS	NUMBER	Number of messages dequeued **Note:** This column will not be incremented until all the subscribers of the message have dequeued the message and its retention time has elapsed.
BROWSED_MSGS	NUMBER	Number of messages that have been browsed
ELAPSED_ENQUEUE_TIME	NUMBER	Total time (in hundredths of a second) spent doing enqueue

Column	Datatype	Description
ELAPSED_DEQUEUE_TIME	NUMBER	Total time (in hundredths of a second) spent doing dequeue
ELAPSED_TRANSFORMATION_TIME	NUMBER	Total time (in hundredths of a second) spent doing transformation
ELAPSED_RULE_EVALUATION_TIME	NUMBER	Total time (in hundredths of a second) spent doing rule evaluation
ENQUEUED_EXPIRY_MSGS	NUMBER	Number of messages enqueued with expiry
ENQUEUED_DELAY_MSGS	NUMBER	Number of messages enqueued with delay
MSGS_MADE_EXPIRED	NUMBER	Number of messages expired by time manager
MSGS_MADE_READY	NUMBER	Number of messages made ready by time manager
LAST_ENQUEUE_TIME	TIMESTAMP(6)	Last message enqueue time
LAST_DEQUEUE_TIME	TIMESTAMP(6)	Last message dequeue time
LAST_TM_EXPIRY_TIME	TIMESTAMP(6)	Last time message was expired by time manager
LAST_TM_READY_TIME	TIMESTAMP(6)	Last time message was made ready by time manager

V$PERSISTENT_SUBSCRIBERS

V$PERSISTENT_SUBSCRIBERS displays information about all active subscribers of the persistent queues in the database. There is one row per instance per queue per subscriber. The rows are deleted when the database (or instance in an Oracle RAC environment) restarts.

Column	Datatype	Description
QUEUE_ID	NUMBER	Identifier for the queue
QUEUE_SCHEMA	VARCHAR2(30)	Owner of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
SUBSCRIBER_ID	NUMBER	Internal subscriber number
SUBSCRIBER_NAME	VARCHAR2(30)	Name of the subscriber
SUBSCRIBER_ADDRESS	VARCHAR2(1024)	Address of the subscribing agent
PROTOCOL	NUMBER	Protocol of the subscribing agent
SUBSCRIBER_TYPE	VARCHAR2(30)	Type of the subscriber: ■ PROXY - Propagation subscriber ■ SUBSCRIBER - Normal subscriber ■ RECIPIENT - Recipient
FIRST_ACTIVITY_TIME	TIMESTAMP(6)	First subscriber activity time since database startup
ENQUEUED_MSGS	NUMBER	Number of messages enqueued since FIRST_ACTIVITY_TIME
DEQUEUED_MSGS	NUMBER	Number of messages dequeued since FIRST_ACTIVITY_TIME
BROWSED_MSGS	NUMBER	Number of messages that have been browsed
EXPIRED_MSGS	NUMBER	Number of messages expired since FIRST_ACTIVITY_TIME
DEQUEUED_MSG_LATENCY	NUMBER	Last dequeued message latency (in seconds)
LAST_ENQUEUE_TIME	TIMESTAMP(6)	Timestamp of the last enqueued message
LAST_DEQUEUE_TIME	TIMESTAMP(6)	Timestamp of the last dequeued message

V$PGA_TARGET_ADVICE

V$PGA_TARGET_ADVICE predicts how the cache hit percentage and over allocation count statistics displayed by the V$PGASTAT performance view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. The prediction is performed for various values of the PGA_AGGREGATE_TARGET parameter, selected around its current value. The advice statistic is generated by simulating the past workload run by the instance.

The content of the view is empty if PGA_AGGREGATE_TARGET is not set. In addition, the content of this view is not updated if the STATISTICS_LEVEL parameter is set to BASIC. Base statistics for this view are reset at instance startup and when the value of the PGA_AGGREGATE_TARGET initialization parameter is dynamically modified.

Column	Datatype	Description
PGA_TARGET_FOR_ESTIMATE	NUMBER	Value of PGA_AGGREGATE_TARGET for this prediction (in bytes)
PGA_TARGET_FACTOR	NUMBER	PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2(3)	Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
BYTES_PROCESSED	NUMBER	Total bytes processed by all the work areas considered by this advice (in bytes)
ESTD_TIME	NUMBER	Time (in seconds) required to process the bytes
ESTD_EXTRA_BYTES_RW	NUMBER	Estimated number of extra bytes which would be read or written if PGA_AGGREGATE_TARGET was set to the value of the PGA_TARGET_FOR_ESTIMATE column. This number is derived from the estimated number and size of work areas which would run in one-pass (or multi-pass) for that value of PGA_AGGREGATE_TARGET.
ESTD_PGA_CACHE_HIT_PERCENTAGE	NUMBER	Estimated value of the cache hit percentage statistic when PGA_AGGREGATE_TARGET equals PGA_TARGET_FOR_ESTIMATE. This column is derived from the above two columns and is equal to BYTES_PROCESSED / (BYTES_PROCESSED + ESTD_EXTRA_BYTES_RW)
ESTD_OVERALLOC_COUNT	NUMBER	Estimated number of PGA memory over-allocations if the value of PGA_AGGREGATE_TARGET is set to PGA_TARGET_FOR_ESTIMATE. A nonzero value means that PGA_TARGET_FOR_ESTIMATE is not large enough to run the work area workload. Hence, the DBA should not set PGA_AGGREGATE_TARGET to PGA_TARGET_FOR_ESTIMATE since Oracle will not be able to honor that target.

See Also: *Oracle Database Performance Tuning Guide* for information on tuning the PGA_AGGREGATE_TARGET initialization parameter using the PGA advice views

V$PGA_TARGET_ADVICE_HISTOGRAM

V$PGA_TARGET_ADVICE_HISTOGRAM predicts how statistics displayed by the V$SQL_WORKAREA_HISTOGRAM dynamic view would be impacted if the value of the PGA_AGGREGATE_TARGET parameter is changed. This prediction is performed for various values of the PGA_AGGREGATE_TARGET parameter, selected around its current value. The advice statistic is generated by simulating the past workload run by the instance.

The content of the view is empty if PGA_AGGREGATE_TARGET is not set. In addition, the content of this view is not updated when the STATISTICS_LEVEL initialization parameter is set to BASIC. Base statistics for this view are reset at instance startup or when the value of the PGA_AGGREGATE_TARGET initialization parameter is dynamically modified.

Column	Datatype	Description
PGA_TARGET_FOR_ESTIMATE	NUMBER	Value of PGA_AGGREGATE_TARGET for this prediction (in bytes)

Column	Datatype	Description
PGA_TARGET_FACTOR	NUMBER	PGA_TARGET_FOR_ESTIMATE / the current value of the PGA_AGGREGATE_TARGET parameter
ADVICE_STATUS	VARCHAR2(3)	Indicates whether the advice is enabled (ON) or disabled (OFF) depending on the value of the STATISTICS_LEVEL parameter
LOW_OPTIMAL_SIZE	NUMBER	Lower bound for the optimal memory requirement of work areas included in this row (in bytes)
HIGH_OPTIMAL_SIZE	NUMBER	Upper bound for the optimal memory requirement of work areas included in this row (in bytes)
ESTD_OPTIMAL_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run optimal given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_ONEPASS_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run one-pass given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_MULTIPASSES_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which are predicted to run multi-pass given a value of PGA_AGGREGATE_TARGET equal to PGA_TARGET_FOR_ESTIMATE
ESTD_TOTAL_EXECUTIONS	NUMBER	Sum of ESTD_OPTIMAL_EXECUTIONS, ESTD_ONEPASS_EXECUTIONS, and ESTD_MULTIPASSES_EXECUTIONS
IGNORED_WORKAREAS_COUNT	NUMBER	Number of work areas with optimal memory requirement between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE ignored in the advice generation due to memory and CPU constraints

See Also: *Oracle Database Performance Tuning Guide* for information on tuning the PGA_AGGREGATE_TARGET initialization parameter using the PGA advice views

V$PGASTAT

V$PGASTAT displays PGA memory usage statistics as well as statistics about the automatic PGA memory manager when it is enabled (that is, when PGA_AGGREGATE_TARGET is set). Cumulative values in V$PGASTAT are accumulated since instance startup.

Column	Datatype	Description
NAME	VARCHAR2(64)	Name of the statistic (see Table 7–4)
VALUE	NUMBER	Statistic value
UNIT	VARCHAR2(12)	Unit for the value: ▪ bytes ▪ microseconds ▪ percent

Table 7–4 V$PGASTAT Statistics

Statistic Name	Description
aggregate PGA target parameter	Current value of the PGA_AGGREGATE_TARGET initialization parameter. If this parameter is not set, then its value is 0 and automatic management of PGA memory is disabled.
aggregate PGA auto target	Amount of PGA memory the Oracle Database can use for work areas running in automatic mode. This amount is dynamically derived from the value of the PGA_AGGREGATE_TARGET initialization parameter and the current work area workload, and continuously adjusted by the Oracle Database. If this value is small compared to the value of PGA_AGGREGATE_TARGET, then a large amount of PGA memory is used by other components of the system (for example, PL/SQL or Java memory) and little is left for work areas. The DBA must ensure that enough PGA memory is left for work areas running in automatic mode.
global memory bound	Maximum size of a work area executed in automatic mode. This value is continuously adjusted by the Oracle Database to reflect the current state of the work area workload. The global memory bound generally decreases when the number of active work areas is increasing in the system. If the value of the global bound decreases below 1 MB, then the value of PGA_AGGREGATE_TARGET should be increased.
total PGA allocated	Current amount of PGA memory allocated by the instance. The Oracle Database attempts to keep this number below the value of the PGA_AGGREGATE_TARGET initialization parameter. However, it is possible for the PGA allocated to exceed that value by a small percentage and for a short period of time when the work area workload is increasing very rapidly or when PGA_AGGREGATE_TARGET is set to a small value.
total PGA used	Indicates how much PGA memory is currently consumed by work areas. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
total PGA used for auto workareas	Indicates how much PGA memory is currently consumed by work areas running under the automatic memory management mode. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
total PGA used for manual workareas	Indicates how much PGA memory is currently consumed by work areas running under the manual memory management mode. This number can be used to determine how much memory is consumed by other consumers of the PGA memory (for example, PL/SQL or Java).
over allocation count	This statistic is cumulative since instance startup. Over allocating PGA memory can happen if the value of PGA_AGGREGATE_TARGET is too small. When this happens, the Oracle Database cannot honor the value of PGA_AGGREGATE_TARGET and extra PGA memory needs to be allocated. If over allocation occurs, then increase the value of PGA_AGGREGATE_TARGET using the information provided by the V$PGA_TARGET_ADVICE view.
bytes processed	Number of bytes processed by memory intensive SQL operators, cumulated since instance startup.
extra bytes read/written	Number of bytes processed during extra passes of the input data, cumulated since instance startup. When a work area cannot run optimal, one or more of these extra passes is performed.
cache hit percentage	A metric computed by the Oracle Database to reflect the performance of the PGA memory component, cumulative since instance startup. A value of 100% means that all work areas executed by the system since instance startup have used an optimal amount of PGA memory. When a work area cannot run optimal, one or more extra passes is performed over the input data. This will reduce the cache hit percentage in proportion to the size of the input data and the number of extra passes performed.

V$PQ_SESSTAT

V$PQ_SESSTAT lists session statistics for parallel queries. After you have run a query or DML operation, you can use the information derived from V$PQ_SESSTAT to view the number of slave processes used, and other information for the session and system.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic: ■ Queries Parallelized - Number of queries run in parallel ■ DDL Parallelized - Number of DDL operations run in parallel ■ DML Parallelized - Number of DML operations run in parallel ■ DFO Trees - Number of executed DFO trees ■ Server Threads - Number of PX servers used ■ Allocation Height - Requested number of servers per instance ■ Allocation Width - Requested number of instances ■ Local Msgs Sent - Number of local (intra-instance) messages sent ■ Distr Msgs Sent - Number of remote (inter-instance) messages sent ■ Local Msgs Recv'd - Number of local (intra-instance) messages received ■ Distr Msgs Recv'd - Number of remote (inter-instance) messages received
LAST_QUERY	NUMBER	Value of the statistic for the last operation
SESSION_TOTAL	NUMBER	Value of the statistic for the entire session to this point in time

V$PQ_SLAVE

V$PQ_SLAVE lists statistics for each of the active parallel execution servers on an instance.

Column	Datatype	Description
SLAVE_NAME	VARCHAR2(4)	Name of the parallel execution server
STATUS	VARCHAR2(4)	Current status of the parallel execution server: ■ BUSY ■ IDLE
SESSIONS	NUMBER	Number of sessions that have used this parallel execution server
IDLE_TIME_CUR	NUMBER	Amount of time spent idle while processing statements in the current session
BUSY_TIME_CUR	NUMBER	Amount of time spent busy while processing statements in the current session
CPU_SECS_CUR	NUMBER	Amount of CPU time spent on the current session
MSGS_SENT_CUR	NUMBER	Number of messages sent while processing statements for the current session
MSGS_RCVD_CUR	NUMBER	Number of messages received while processing statements for the current session
IDLE_TIME_TOTAL	NUMBER	Total amount of time this query server has been idle
BUSY_TIME_TOTAL	NUMBER	Total amount of time this query server has been active
CPU_SECS_TOTAL	NUMBER	Total amount of CPU time this query server has used to process statements
MSGS_SENT_TOTAL	NUMBER	Total number of messages this query server has sent
MSGS_RCVD_TOTAL	NUMBER	Total number of messages this query server has received

V$PQ_SYSSTAT

V$PQ_SYSSTAT lists system statistics for parallel queries. After you have run a query or DML operation, you can use the information derived from V$PQ_SYSSTAT to view the number of slave processes used, and other information for the system.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic: ■ Servers Busy - Number of currently busy servers on this instance ■ Servers Idle - Number of currently idle servers on this instance ■ Servers Highwater - Number of active servers on this instance that have partaken in >= 1 operation so far ■ Server Sessions - Total number of operations executed in all servers on this instance ■ Servers Started - Total number of servers started on this instance ■ Servers Shutdown - Total number of servers shutdown on this instance ■ Servers Cleaned Up - Total number of servers on this instance cleaned up due to process death ■ Queries Initiated - Total number of parallel queries initiated on this instance ■ DDL Initiated - Total number of parallel DDL operations that were initiated ■ DML Initiated - Total number of parallel DML operations that were initiated ■ DFO Trees - Total number of DFO trees executed on this instance ■ Local Msgs Sent - Total number of local (intra-instance) messages sent on this instance ■ Distr Msgs Sent - Total number of remote (inter-instance) messages sent on this instance ■ Local Msgs Recv'd - Total number of remote (inter-instance) messages received on this instance ■ Distr Msgs Recv'd - Total number of remote (inter-instance) messages received on this instance
VALUE	NUMBER	Value of the statistic

V$PQ_TQSTAT

V$PQ_TQSTAT contains statistics on parallel execution operations. The statistics are compiled after the query completes and only remain for the duration of the session. It displays the number of rows processed through each parallel execution server at each stage of the execution tree. This view can help determine skew problems in a query's execution. (Note that for PDML, information from V$PQ_TQSTAT is available only after a commit or rollback operation.)

Column	Datatype	Description
DFO_NUMBER	NUMBER	Data flow operator (DFO) tree number to differentiate queries
TQ_ID	NUMBER	Table queue ID within the query, which represents the connection between two DFO nodes in the query execution tree
SERVER_TYPE	VARCHAR2(10)	The role in table queue - producer/consumer/ranger
NUM_ROWS	NUMBER	The number of rows produced/consumed
BYTES	NUMBER	The number of bytes produced/consumed
OPEN_TIME	NUMBER	Time (seconds) the table queue remained open
AVG_LATENCY	NUMBER	Time (minutes) for a message to be dequeued after it enters the queue
WAITS	NUMBER	The number of waits encountered during dequeue
TIMEOUTS	NUMBER	The number of timeouts when waiting for a message
PROCESS	VARCHAR2(10)	Process ID
INSTANCE	NUMBER	Instance ID

V$PROCESS

V$PROCESS displays information about the currently active processes. While the LATCHWAIT column indicates what latch a process is waiting for, the LATCHSPIN column indicates what latch a process is spinning on. On multi-processor machines, Oracle processes will spin on a latch before waiting on it.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the process state object
PID	NUMBER	Oracle process identifier
SPID	VARCHAR2(24)	Operating system process identifier
USERNAME	VARCHAR2(15)	Operating system process username **Note:** Any two-task user coming across the network has "-T" appended to the username.
SERIAL#	NUMBER	Process serial number
TERMINAL	VARCHAR2(30)	Operating system terminal identifier
PROGRAM	VARCHAR2(48)	Program in progress
TRACEID	VARCHAR2(255)	Trace file identifier
TRACEFILE	VARCHAR2(513)	Trace file name of the process
BACKGROUND	VARCHAR2(1)	1 for a background process; NULL for a normal process
LATCHWAIT	VARCHAR2(8)	Address of the latch the process is waiting for; NULL if none
LATCHSPIN	VARCHAR2(8)	Address of the latch the process is spinning on; NULL if none
PGA_USED_MEM	NUMBER	PGA memory currently used by the process
PGA_ALLOC_MEM	NUMBER	PGA memory currently allocated by the process (including free PGA memory not yet released to the operating system by the server process)
PGA_FREEABLE_MEM	NUMBER	Allocated PGA memory which can be freed
PGA_MAX_MEM	NUMBER	Maximum PGA memory ever allocated by the process

V$PROCESS_MEMORY

V$PROCESS_MEMORY displays dynamic PGA memory usage by named component categories for each process.

Column	Datatype	Description
PID	NUMBER	Oracle process identifier
SERIAL#	NUMBER	Oracle process serial number
CATEGORY	VARCHAR2(15)	Category name. Categories include "SQL", "PL/SQL", "OLAP" and "JAVA". Special categories are "Freeable" and "Other". Freeable memory has been allocated to the process by the operating system, but has not been allocated to a category. "Other" memory has been allocated to a category, but not to one of the named categories.
ALLOCATED	NUMBER	Bytes of PGA memory allocated by the process for the category. For the "Freeable" category, it is the amount of free PGA memory eligible to be released to the operating system.
USED	NUMBER	Bytes of PGA memory used by the process for the category. For "Freeable", the value is zero. For "Other", the value is NULL for performance reasons.
MAX_ALLOCATED	NUMBER	Maximum bytes of PGA memory ever allocated by the process for the category.

V$PROPAGATION_RECEIVER

V$PROPAGATION_RECEIVER displays information about buffer queue propagation schedules on the receiving (destination) side. The values are reset to zero when the database (or instance in a Real Application Clusters (RAC) environment) restarts, when propagation migrates to another instance, or when an unscheduled propagation is attempted.

Column	Datatype	Description
SRC_QUEUE_SCHEMA	VARCHAR2(30)	Name of the source schema
SRC_QUEUE_NAME	VARCHAR2(30)	Name of the source queue
SRC_DBNAME	VARCHAR2(128)	Name of the source database
DST_QUEUE_SCHEMA	VARCHAR2(30)	Name of the destination schema
DST_QUEUE_NAME	VARCHAR2(30)	Name of the destination queue
STARTUP_TIME	DATE	Startup time of this schedule. This time changes when the source/destination database gets restarted.
HIGH_WATER_MARK	NUMBER	High-watermark of the messages received
ACKNOWLEDGEMENT	NUMBER	Acknowledgement of the messages received by the receiver
LAST_RECEIVED_MSG	NUMBER	Last received message
TOTAL_MSGS	NUMBER	Total number of messages
ELAPSED_UNPICKLE_TIME	NUMBER	Elapsed unpickle time
ELAPSED_RULE_TIME	NUMBER	Elapsed rule time
ELAPSED_ENQUEUE_TIME	NUMBER	Elapsed enqueue time

V$PROPAGATION_SENDER

V$PROPAGATION_SENDER displays information about buffer queue propagation schedules on the sending (source) side. The values are reset to zero when the database (or instance in a Real Application Clusters (RAC) environment) restarts, when propagation migrates to another instance, or when an unscheduled propagation is attempted.

Column	Datatype	Description
QUEUE_ID	NUMBER	Queue identifier of the queue
QUEUE_SCHEMA	VARCHAR2(30)	Schema of the queue
QUEUE_NAME	VARCHAR2(30)	Name of the queue
DST_QUEUE_SCHEMA	VARCHAR2(30)	Destination schema of the queue
DST_QUEUE_NAME	VARCHAR2(30)	Name of the destination queue
STARTUP_TIME	DATE	Time at which the propagation started
DBLINK	VARCHAR2(128)	Name of the destination database link
HIGH_WATER_MARK	NUMBER	High-watermark of the messages sent
ACKNOWLEDGEMENT	NUMBER	Acknowledgement of the messages received by the receiver
SCHEDULE_STATUS	VARCHAR2(30)	Status of the propagation schedule
TOTAL_MSGS	NUMBER	Total messages propagated
TOTAL_BYTES	NUMBER	Total bytes propagated
ELAPSED_DEQUEUE_TIME	NUMBER	Elapsed dequeue time (in hundredths of a second)
ELAPSED_PICKLE_TIME	NUMBER	Elapsed pickle time (time taken to linearize a logical change record (LCR) into a stream of bytes that can be sent over the network) (in hundredths of a second)

Column	Datatype	Description
ELAPSED_PROPAGATION_TIME	NUMBER	Elapsed propagation time (in hundredths of a second)
MAX_NUM_PER_WIN	NUMBER	Maximum bytes per window
MAX_SIZE	NUMBER	Maximum bytes sent per window
LAST_MSG_LATENCY	NUMBER	Last propagated message latency
LAST_MSG_ENQUEUE_TIME	TIMESTAMP(6)	Last propagated message enqueue time
LAST_MSG_PROPAGATION_TIME	TIMESTAMP(6)	Last time when the message was propagated
LAST_LCR_LATENCY	NUMBER	Last propagated LCR latency
LAST_LCR_CREATION_TIME	DATE	Last propagated LCR timestamp
LAST_LCR_PROPAGATION_TIME	DATE	Last time when the LCR was propagated
DST_DATABASE_NAME	VARCHAR2(128)	Global name of the destination database

V$PROXY_ARCHIVEDLOG

V$PROXY_ARCHIVEDLOG contains descriptions of archived log backups that were taken using the proxy copy functionality.

In a proxy copy, the media manager takes over the operations of backing up and restoring data. Each row represents a backup of one control file.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record identifier
STAMP	NUMBER	Proxy copy stamp
DEVICE_TYPE	VARCHAR2(17)	Type of media device that stores the proxy copy
HANDLE	VARCHAR2(513)	Name or "handle" for the proxy copy
COMMENTS	VARCHAR2(81)	Comments about the proxy copy
MEDIA	VARCHAR2(65)	A comment that contains further information about the media manager that created this backup
MEDIA_POOL	NUMBER	Number of the media pool in which the proxy copy is stored
TAG	VARCHAR2(32)	Tag for the proxy copy
STATUS	VARCHAR2(1)	Status of the backup set: ■ A - Available ■ U - Unavailable ■ X - Expired ■ D - Deleted
DELETED	VARCHAR2(3)	Indicates whether this record has been deleted (YES) or not (NO)
THREAD#	NUMBER	Number of the redo thread
SEQUENCE#	NUMBER	Log sequence number
RESETLOGS_CHANGE#	NUMBER	RESETLOGS SCN of the database incarnation to which this archived log belongs
RESETLOGS_TIME	DATE	RESETLOGS time stamp of the database incarnation to which this archived log belongs
FIRST_CHANGE#	NUMBER	First SCN of this redo log
FIRST_TIME	DATE	Time when Oracle switched into the redo log
NEXT_CHANGE#	NUMBER	First SCN of the next redo log in the thread

Column	Datatype	Description
NEXT_TIME	DATE	First time stamp of the next redo log in the thread
BLOCKS	NUMBER	Size of this archived redo log (in operating system blocks)
BLOCK_SIZE	NUMBER	Block size for the copy (in bytes)
START_TIME	DATE	Time when the proxy copy was initiated
COMPLETION_TIME	DATE	Time when the proxy copy was completed
ELAPSED_SECONDS	NUMBER	Duration of the proxy copy
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS stamp
TERMINAL	VARCHAR2(3)	Indicates whether this record corresponds to a terminal archived redo log, as defined in V$ARCHIVED_LOG (YES) or not (NO)

V$PROXY_ARCHIVELOG_DETAILS

V$PROXY_ARCHIVELOG_DETAILS contains information about all available archive log proxy copies.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
COPY_KEY	NUMBER	Copy identifier
THREAD#	NUMBER	Redo thread number
SEQUENCE#	NUMBER	Redo log sequence number
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the database when this log was written
RESETLOGS_TIME	DATE	Resetlogs time of the database when this log was written
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.
TAG	VARCHAR2(32)	Proxy copy tag
FIRST_CHANGE#	NUMBER	First change number in the archived log
NEXT_CHANGE#	NUMBER	First change number in the next log
FIRST_TIME	DATE	Timestamp of the first change
NEXT_TIME	DATE	Timestamp of the next change
OUTPUT_BYTES	NUMBER	Total output bytes written
COMPLETION_TIME	DATE	Completion time
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$PROXY_ARCHIVELOG_SUMMARY

V$PROXY_ARCHIVELOG_SUMMARY provides summary information about the output proxy archive log file.

Column	Datatype	Description
NUM_FILES_BACKED	NUMBER	Number of archived log files backed up

Column	Datatype	Description
NUM_DISTINCT_FILES_BACKED	NUMBER	Number of distinct archived log files backed up
MIN_FIRST_CHANGE#	NUMBER	Minimum first change SCN
MAX_NEXT_CHANGE#	NUMBER	Maximum first change SCN
MIN_FIRST_TIME	DATE	Minimum first change time. Forms the redo range, along with MAX_NEXT_TIME.
MAX_NEXT_TIME	DATE	Maximum next change time
OUTPUT_BYTES	NUMBER	Total output size, in bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$PROXY_COPY_DETAILS

V$PROXY_COPY_DETAILS contains information about all available control file and datafile proxy copies.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session recid
SESSION_STAMP	NUMBER	Session stamp
COPY_KEY	NUMBER	Copy identifier
FILE#	NUMBER	Absolute datafile number, or 0 if this is a control file backup
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.
TAG	VARCHAR2(32)	Proxy copy tag
CREATION_CHANGE#	NUMBER	Datafile creation change number
CREATION_TIME	DATE	Datafile creation timestamp
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
OUTPUT_BYTES	NUMBER	Total output bytes written
COMPLETION_TIME	DATE	Completion time
CONTROLFILE_TYPE	VARCHAR2(1)	Type of control file: ■ B - Normal control file ■ S - Standby control file
KEEP	VARCHAR2(3)	Indicates whether this backup set has a retention policy that is different than the value for the configure retention policy (YES) or not (NO)
KEEP_UNTIL	DATE	If specified, then this is the date after which the backup becomes obsolete. If this column is NULL, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(11)	Additional retention options for this backup set: ■ LOGS - Indicates a long-term backup made with the LOGS keyword, which is now deprecated ■ BACKUP_LOGS - Indicates that the backup was made in open mode, so archived log backups must be applied to make it consistent ■ NOLOGS - Indicates a consistent backup made when the database was mounted. ■ NULL - Indicates that this backup has no KEEP options and becomes obsolete based on the retention policy

Column	Datatype	Description
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$PROXY_COPY_SUMMARY

V$PROXY_COPY_SUMMARY provides summary information about the output proxy datafile and control file.

Column	Datatype	Description
NUM_COPIES	NUMBER	Number of copies created
NUM_DISTINCT_COPIES	NUMBER	Number of distinct copies (that contain datafiles with different checkpoints)
MIN_CHECKPOINT_CHANGE#	NUMBER	Minimum checkpoint change SCN
MAX_CHECKPOINT_CHANGE#	NUMBER	Maximum checkpoint change SCN
MIN_CHECKPOINT_TIME	DATE	Minimum checkpoint change time
MAX_CHECKPOINT_TIME	DATE	Maximum checkpoint change time
OUTPUT_BYTES	NUMBER	Total output bytes
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Displayable format for output bytes

V$PROXY_DATAFILE

V$PROXY_DATAFILE contains descriptions of datafile and control file backups that are taken with Proxy Copy. Each row represents a backup of one database file.

Column	Datatype	Description
RECID	NUMBER	Proxy copy record ID
STAMP	NUMBER	Proxy copy record stamp
DEVICE_TYPE	VARCHAR2(17)	Type of the device on which the copy resides
HANDLE	VARCHAR2(513)	Proxy copy handle identifies the copy for restore
COMMENTS	VARCHAR2(81)	Comment returned by the operating system or storage subsystem. This value is informational only. It is not needed for restore.
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command
TAG	VARCHAR2(32)	Proxy copy tag
STATUS	VARCHAR2(1)	Status of the backup set: ■ A - Available ■ U - Unavailable ■ X - Expired ■ D - Deleted
DELETED	VARCHAR2(3)	Indicates whether this record has been deleted (YES) or not (NO)
FILE#	NUMBER	Absolute datafile number, or 0 if this is a control file backup
CREATION_CHANGE#	NUMBER	Datafile creation change number
CREATION_TIME	DATE	Datafile creation Timestamp
RESETLOGS_CHANGE#	NUMBER	Resetlogs change number of the datafile when the copy was made
RESETLOGS_TIME	DATE	Resetlogs timestamp of the datafile when the copy was made

Column	Datatype	Description
CHECKPOINT_CHANGE#	NUMBER	Checkpoint change number of the datafile when the copy was made
CHECKPOINT_TIME	DATE	Checkpoint timestamp of the datafile when the copy was made
ABSOLUTE_FUZZY_CHANGE#	NUMBER	Highest change in any block of the file, if known
RECOVERY_FUZZY_CHANGE#	NUMBER	Highest change written to the file by media recovery
RECOVERY_FUZZY_TIME	DATE	Timestamp of the highest change written to the file by media recovery
INCREMENTAL_LEVEL	NUMBER	If this backup is part of an incremental backup strategy, then 0. Otherwise null.
ONLINE_FUZZY	VARCHAR2(3)	Indicates whether this copy was made after a crash or offline immediate (or is a copy of a copy which was taken improperly while the database was open) (YES) or not (NO). Recovery will need to apply all redo up to the next crash recovery marker to make the file consistent.
BACKUP_FUZZY	VARCHAR2(3)	Indicates whether this is a copy taken using the BEGIN BACKUP | END BACKUP technique (YES) or not (NO). The BEGIN BACKUP | END BACKUP technique is used internally when proxy copies of open files are created. Recovery will need to apply all redo up to the end backup marker to make this copy consistent.
BLOCKS	NUMBER	Size of the copy (in blocks). Also the size of the datafile when the copy was made.
BLOCK_SIZE	NUMBER	Block size of the datafile
OLDEST_OFFLINE_RANGE	NUMBER	If the file number is 0 (that is, this is a control file backup), the RECID of the oldest offline range record in this control file copy. 0 for datafile copies.
START_TIME	DATE	Starting time
COMPLETION_TIME	DATE	Completion time
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
CONTROLFILE_TYPE	VARCHAR2(1)	Type of control file: ■ B - Normal control file ■ S - Standby control file
KEEP	VARCHAR2(3)	Indicates whether this backup set has a retention policy that is different than the value for the configure retention policy (YES) or not (NO)
KEEP_UNTIL	DATE	If specified, then this is the date after which the backup becomes obsolete. If this column is NULL, then the backup never expires.
KEEP_OPTIONS	VARCHAR2(11)	Additional retention options for this backup set: ■ LOGS - Indicates a long-term backup made with the LOGS keyword, which is now deprecated ■ BACKUP_LOGS - Indicates that the backup was made in open mode, so archived log backups must be applied to make it consistent ■ NOLOGS - Indicates a consistent backup made when the database was mounted. ■ NULL - Indicates that this backup has no KEEP options and becomes obsolete based on the retention policy
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS stamp
FOREIGN_DBID	NUMBER	Foreign DBID of the database from which this datafile was transported. The value is 0 if the file backed up is not a foreign database file.
PLUGGED_READONLY	VARCHAR2(3)	Indicates whether this is a proxy copy of a transported read-only foreign file (YES) or not (NO)
PLUGIN_CHANGE#	NUMBER	SCN at which the foreign datafile was transported into the database. The value is 0 if this file is not a foreign database file.
PLUGIN_RESETLOGS_CHANGE#	NUMBER	SCN of the RESETLOGS operation for the incarnation into which this foreign file was transported. The value is 0 if this file is not a foreign database file.

Column	Datatype	Description
PLUGIN_RESETLOGS_TIME	DATE	Time of the RESETLOGS operation for the incarnation into which this foreign file was transported. The value is 0 if this file is not a foreign database file.

V$PWFILE_USERS

V$PWFILE_USERS lists users who have been granted SYSDBA and SYSOPER privileges as derived from the password file.

Column	Datatype	Description
USERNAME	VARCHAR2(30)	Name of the user that is contained in the password file
SYSDBA	VARCHAR2(5)	Indicates whether the user can connect with SYSDBA privileges (TRUE) or not (FALSE)
SYSOPER	VARCHAR2(5)	Indicates whether the user can connect with SYSOPER privileges (TRUE) or not (FALSE)
SYSASM	VARCHAR2(5)	Indicates whether the user can connect with SYSASM privileges (TRUE) or not (FALSE)

V$PX_INSTANCE_GROUP

V$PX_INSTANCE_GROUP provides information about the instance groups being used for parallel operations by the current session.

Column	Datatype	Description
QC_INSTANCE_GROUP	VARCHAR2(64)	The instance group being used by this session for parallel operations. This value repeats for every row returned.
WHY	VARCHAR2(23)	Where the current instance group name comes from, as follows: SERVICE - the instance group being used is from the session's service name PARALLEL_INSTANCE_GROUP - the instance group being used is being used because the PARALLEL_INSTANCE_GROUP initialization parameter has been set. The value repeats for every row returned.
INSTANCE_NUMBER	NUMBER	Instance number of the instance providing this instance group. There will be one row for each instance in the instance group that the Query Coordinator is using.

V$PX_PROCESS

V$PX_PROCESS contains information about the sessions running parallel execution.

Column	Datatype	Description
SERVER_NAME	VARCHAR2(4)	The name of the PX server (P000, P001, and so on)
STATUS	VARCHAR2(9)	The state of the PX server (IN USE|AVAILABLE)
PID	NUMBER	The process identifier
SPID	VARCHAR2(12)	OS process ID
SID	NUMBER	The session ID of the PX server, if in use
SERIAL#	NUMBER	The session serial number of the PX server, if in use

V$PX_PROCESS_SYSSTAT

V$PX_PROCESS_SYSSTAT contains information about the sessions running parallel execution.

Column	Datatype	Description
STATISTIC	VARCHAR2(30)	Name of the statistic: - Servers In Use - Number of PX servers currently performing parallel operations - Servers Available - Number of PX servers available to perform parallel operations - Servers Started - Number of times the system has had to create a PX server process - Servers Shutdown - Number of times a PX server process has been shut down. A PX server process will be shut down if it has not been used recently. If this value is large, then consider increasing the parameter. This will improve performance by avoiding the latency of PX server process creation. - Servers HWM - Maximum number of concurrent PX server processes If this number is equal to the PARALLEL_MAX_SERVERS initialization parameter, then consider increasing the parameter. This could allow you to increase your throughput, especially if your system is under-utilized and the V$SYSSTAT statistic "Parallel operations downgraded to serial" is large. - Servers Cleaned Up - Number of times PMON had to clean up a PX server. This should only happen during abnormal termination of a parallel operation. If this number is large, then you should determine the cause. - Server Sessions - Total number of sessions created by all PX servers - Memory Chunks Allocated - Number of large memory chunks allocated by PX servers - Memory Chunks Freed - Number of large memory chunks freed - Memory Chunks Current - Number of large memory chunks currently being used - Memory Chunks HWM - Maximum number of concurrently allocated chunks - Buffers allocated - Number of times a message buffer has been allocated - Buffers freed - Number of times a message buffer has been freed - Buffers Current - Number of message buffers currently being used - Buffers HWM - Maximum number of concurrently allocated message buffers
VALUE	NUMBER	Value of the statistic

V$PX_SESSION

V$PX_SESSION contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4 | 8)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator

Column	Datatype	Description
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this cluster database process belongs
SERVER_SET	NUMBER	The logical set of servers to which this cluster database process belongs. A single server group will have at most two server sets.
SERVER#	NUMBER	The logical number of the cluster database process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions

V$PX_SESSTAT

V$PX_SESSTAT contains information about the sessions running parallel execution.

Column	Datatype	Description
SADDR	RAW(4 | 8)	Session address
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Session serial number
QCSID	NUMBER	Session identifier of the parallel coordinator
QCSERIAL#	NUMBER	Session serial number of the parallel coordinator
QCINST_ID	NUMBER	Instance number on which the parallel coordinator is running
SERVER_GROUP	NUMBER	The logical group of servers to which this cluster database process belongs
SERVER_SET	NUMBER	The logical set of servers that this cluster database process belongs to. A single server group will have at most two server sets.
SERVER#	NUMBER	The logical number of the cluster database process within a server set
DEGREE	NUMBER	The degree of parallelism being used by the server set
REQ_DEGREE	NUMBER	The degree of parallelism that was requested by the user when the statement was issued and prior to any resource, multi-user, or load balancing reductions
STATISTIC#	NUMBER	Statistic number
VALUE	NUMBER	Statistic value

V$QUEUE

V$QUEUE contains information on the shared server message queues.

Column	Datatype	Description
PADDR	RAW(4 | 8)	Address of the process that owns the queue
TYPE	VARCHAR2(10)	Type of queue: ▪ COMMON - Processed by servers ▪ DISPATCHER
QUEUED	NUMBER	Number of items in the queue
WAIT	NUMBER	Total time that all items in this queue have waited (in hundredths of a second). Divide by TOTALQ for average wait per item.
TOTALQ	NUMBER	Total number of items that have ever been in the queue

V$QUEUEING_MTH

V$QUEUEING_MTH displays all available queuing resource allocation methods.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the queuing resource allocation method

V$RECOVER_FILE

V$RECOVER_FILE displays the status of files needing media recovery.

Column	Datatype	Description
FILE#	NUMBER	File identifier number
ONLINE	VARCHAR2(7)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in ONLINE_STATUS.
ONLINE_STATUS	VARCHAR2(7)	Online status (ONLINE, OFFLINE)
ERROR	VARCHAR2(18)	Why the file needs to be recovered: NULL if reason unknown, or OFFLINE NORMAL if recovery not needed
CHANGE#	NUMBER	SCN where recovery must start
TIME	DATE	Time of SCN when recovery must start

V$RECOVERY_FILE_DEST

V$RECOVERY_FILE_DEST displays information about the disk quota and current disk usage in the flash recovery area.

Column	Datatype	Description
NAME	VARCHAR2(513)	Location name. This is the value specified in the DB_RECOVERY_FILE_DEST initialization parameter.
SPACE_LIMIT	NUMBER	Maximum amount of disk space (in bytes) that the database can use for the flash recovery area. This is the value specified in the DB_RECOVERY_FILE_DEST_SIZE initialization parameter.
SPACE_USED	NUMBER	Amount of disk space (in bytes) used by flash recovery area files created in current and all previous flash recovery areas. Changing flash recovery areas does not reset SPACE_USED to 0.
SPACE_RECLAIMABLE	NUMBER	Total amount of disk space (in bytes) that can be created by deleting obsolete, redundant, and other low priority files from the flash recovery area
NUMBER_OF_FILES	NUMBER	Number of files in the flash recovery area

V$RECOVERY_FILE_STATUS

V$RECOVERY_FILE_STATUS contains one row for each datafile for each RECOVER statement. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_FILE_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
FILENUM	NUMBER	Number of the file being recovered
FILENAME	VARCHAR2(513)	Filename of the datafile being recovered

Column	Datatype	Description
STATUS	VARCHAR2(13)	Status of the recovery: ■ IN RECOVERY ■ CURRENT ■ NOT RECOVERED

See Also: *Oracle Database Backup and Recovery Basics*

V$RECOVERY_LOG

V$RECOVERY_LOG lists information about archived logs that are needed to complete media recovery. This information is derived from the log history view, V$LOG_HISTORY.

V$RECOVERY_LOG contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_LOG will be empty to all other Oracle users.

Column	Datatype	Description
THREAD#	NUMBER	Thread number of the archived log
SEQUENCE#	NUMBER	Sequence number of the archived log
TIME	DATE	Time of the first entry (lowest SCN) in the log
ARCHIVE_NAME	VARCHAR2(513)	Name of the file when archived, using the naming convention specified by the LOG_ARCHIVE_FORMAT initialization parameter **See Also:** "LOG_ARCHIVE_FORMAT" on page 1-86

See Also: "V$LOG_HISTORY" on page 7-16 and *Oracle Database Backup and Recovery User's Guide*

V$RECOVERY_PROGRESS

V$RECOVERY_PROGRESS can be used to track database recovery operations to ensure that they are not stalled, and also to estimate the time required to complete the operation in progress.

V$RECOVERY_PROGRESS is a subview of V$SESSION_LONGOPS.

Column	Datatype	Description
START_TIME	DATE	Start time of the recovery operation
TYPE	VARCHAR2(64)	Type of recovery operation being performed
ITEM	VARCHAR2(32)	Item being measured
UNITS	VARCHAR2(32)	The units of measurement for each item
SOFAR	NUMBER	Amount of work done so far
TOTAL	NUMBER	Total amount of work expected
TIMESTAMP	DATE	Timestamp of the last redo record applied

See Also: *Oracle Database Backup and Recovery User's Guide*

V$RECOVERY_STATUS

V$RECOVERY_STATUS contains statistics of the current recovery process. This view contains useful information only for the Oracle process doing the recovery. When Recovery Manager directs a server process to perform recovery, only Recovery Manager is able to view the relevant information in this view. V$RECOVERY_STATUS will be empty to all other Oracle users.

Column	Datatype	Description
RECOVERY_CHECKPOINT	DATE	Point in time to which the recovery has occurred. If no logs have been applied, this is the point in time the recovery starts.
THREAD	NUMBER	Number of the redo thread currently being processed
SEQUENCE_NEEDED	NUMBER	Log sequence number of the log needed by the recovery process. The value is 0 if no log is needed.
SCN_NEEDED	VARCHAR2(16)	Low SCN of the log needed by recovery. The value is 0 if unknown or no log is needed.
TIME_NEEDED	DATE	Time when the log was created. The value is midnight on 1/1/88 if the time is unknown or if no log is needed.
PREVIOUS_LOG_NAME	VARCHAR2(513)	Filename of the log
PREVIOUS_LOG_STATUS	VARCHAR2(13)	Status of the previous log. Contains one of the following values: RELEASE; WRONG NAME; MISSING NAME; UNNEEDED NAME; NONE
REASON	VARCHAR2(13)	Reason recovery is returning control to the user (NEED LOG|LOG REUSED|THREAD DISABLED)

See Also: *Oracle Database Backup and Recovery Basics*

V$REDO_DEST_RESP_HISTOGRAM

V$REDO_DEST_RESP_HISTOGRAM provides statistical information for each redo transport destination.

Column	Datatype	Description
DEST_ID	NUMBER	A non-negative integer value from 1 - 10 for each possible LGWR SYNC standby destination
TIME	VARCHAR2(20)	A text string that shows the last wall-clock time that a bucket was hit
DURATION	NUMBER	A positive integer value that represents a bucket of seconds, 1, 2, 3, up to 300 seconds, followed by 5 additional buckets that represent 600, 1200, 2400, 4800, and 9600 (>= 4801) seconds
FREQUENCY	NUMBER	A non-negative integer that shows the number of times a particular bucket was hit by the SYNC LNS process

V$REPLPROP

V$REPLPROP displays information about the parallel propagation currently in progress at the replication site. Use this view to determine which transactions are currently being propagated, the number of calls propagated in each transaction, and the current activity of the parallel propagation slave processes or parallel propagation coordinator process.

Column	Datatype	Description
SID	NUMBER	Session identifier

Column	Datatype	Description
SERIAL#	NUMBER	Session serial number. Used to identify uniquely a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends with, and another session begins with, the same session ID.
NAME	VARCHAR2(71)	Name of the propagation session: ■ Replication Parallel Prop Slave*n* - Slave process *n* is active, either waiting, pushing deferred transactions, purging metadata, or creating an error transaction ■ Replication Parallel Prop Coordinator - Coordinator process is active, and either waiting, sleeping, or scheduling slaves to perform operations The coordinator process reads transactions from the deferred transaction queue and assigns them to the slave processes. Then, the slaves propagate the transactions to the destination site. When the slaves push transactions in a push session, the slaves remain active until the push session completes, even if there are no more transactions to push.
DBLINK	VARCHAR2(128)	Database link on which this replication session is propagating
STATE	VARCHAR2(12)	State of the propagation session: ■ WAIT - Slave processes or the coordinator process is waiting for an event (that is, a message) ■ SLEEP - Coordinator process is sleeping for the duration of the delay_seconds setting. **Note:** delay_seconds is set with the DBMS_DEFER_SYS.SCHEDULE_PUSH procedure. ■ PUSH - Slave processes are pushing transactions from the deferred transaction queue to the remote site ■ PURGE - Slave processes are purging metadata related to successfully applied transactions from the remote site ■ CREATE ERROR - Slave processes are creating an error transaction. In this case, an error or a conflict occurred while the slaves were pushing deferred transactions to the remote site. ■ SCHEDULE TXN - Coordinator process is determining the order that transactions are applied and assigning slave processes to execute the transactions
XID	VARCHAR2(22)	If the session is a slave session, then indicates the transaction id of the transaction that the slave is currently propagating.
SEQUENCE	NUMBER	If the process is a slave process, then indicates the sequence number of the calls propagated in the current operation, if relevant. Each transaction must process one or more calls, and the value of SEQUENCE starts at zero and increases as each call is processed. The SEQUENCE value shows the call that is currently being processed in each transaction. This value increases until the slave has processed all of the calls in a transaction.

Note: This view only contains data when deferred transactions are being pushed using parallel propagation at the current site. The parallelism parameter must be set to 1 or higher in the DBMS_DEFER_SYS.PUSH function for a push to use parallel propagation. Otherwise, the push uses serial propagation, and no data appears in this view during the push.

V$REPLQUEUE

V$REPLQUEUE displays statistics about the replication deferred transactions queue. All values are stored since the start of the current database instance.

Column	Datatype	Description
TXNS_ENQUEUED	NUMBER	Number of transactions enqueued in the deferred transactions queue
CALLS_ENQUEUED	NUMBER	Number of calls enqueued into the deferred transactions queue
TXNS_PURGED	NUMBER	Number of transactions purged from the deferred transactions queue
LAST_ENQUEUE_TIME	DATE	Date when the last transaction was enqueued into the deferred transaction queue (null if no transactions have been enqueued into the deferred transaction queue since the instance started)
LAST_PURGE_TIME	DATE	Date when the last transaction was purged from the deferred transaction queue (null if no transactions have been purged from the deferred transaction queue since the instance started)

V$REQDIST

V$REQDIST lists statistics for the histogram of shared server dispatcher request times, divided into 12 buckets, or ranges of time. The time ranges grow exponentially as a function of the bucket number.

Column	Datatype	Description
BUCKET	NUMBER	Bucket number: 0 - 11; the maximum time for each bucket is (4 * 2^N)/100 seconds
COUNT	NUMBER	Count of requests whose total time to complete (excluding wait time) falls in this range

V$RESERVED_WORDS

V$RESERVED_WORDS displays a list of all SQL keywords. To determine whether a particular keyword is reserved in any way, check the RESERVED, RES_TYPE, RES_ATTR, and RES_SEMI columns.

Column	Datatype	Description
KEYWORD	VARCHAR2(30)	Name of the keyword
LENGTH	NUMBER	Length of the keyword
RESERVED	VARCHAR2(1)	Indicates whether the keyword cannot be used as an identifier (Y) or whether the keyword is not reserved (N)
RES_TYPE	VARCHAR2(1)	Indicates whether the keyword cannot be used as a type name (Y) or whether the keyword is not reserved (N)
RES_ATTR	VARCHAR2(1)	Indicates whether the keyword cannot be used as an attribute name (Y) or whether the keyword is not reserved (N)
RES_SEMI	VARCHAR2(1)	Indicates whether the keyword is not allowed as an identifier in certain situations, such as in DML (Y) or whether the keyword is not reserved (N)
DUPLICATE	VARCHAR2(1)	Indicates whether the keyword is a duplicate of another keyword (Y) or whether the keyword is not a duplicate (N)

V$RESOURCE

V$RESOURCE contains resource name and address information.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the resource object
TYPE	VARCHAR2(2)	Resource type; the resource types are listed in Table 7–1 on page 7-14
ID1	NUMBER	Resource identifier #1

Column	Datatype	Description
ID2	NUMBER	Resource identifier #2

V$RESOURCE_LIMIT

V$RESOURCE_LIMIT displays information about global resource use for some of the system resources. Use this view to monitor the consumption of resources so that you can take corrective action, if necessary. Many of the resources correspond to initialization parameters listed in Table 7–5.

Some resources, those used by DLM for example, have an initial allocation (soft limit), and the hard limit, which is theoretically infinite (although in practice it is limited by SGA size). During SGA reservation/initialization, a place is reserved in SGA for the INITIAL_ALLOCATION of resources, but if this allocation is exceeded, additional resources are allocated up to the value indicated by LIMIT_VALUE. The CURRENT_UTILIZATION column indicates whether the initial allocation has been exceeded. When the initial allocation value is exceeded, the additional required resources are allocated from the shared pool, where they must compete for space with other resources.

A good choice for the value of INITIAL_ALLOCATION will avoid the contention for space. For most resources, the value for INITIAL_ALLOCATION is the same as the LIMIT_VALUE. Exceeding LIMIT_VALUE results in an error.

Column	Datatype	Description
RESOURCE_NAME	VARCHAR2(30)	Name of the resource (see Table 7–5)
CURRENT_UTILIZATION	NUMBER	Number of (resources, locks, or processes) currently being used
MAX_UTILIZATION	NUMBER	Maximum consumption of this resource since the last instance start-up
INITIAL_ALLOCATION	VARCHAR2(10)	Initial allocation. This will be equal to the value specified for the resource in the initialization parameter file (UNLIMITED for infinite allocation).
LIMIT_VALUE	VARCHAR2(10)	Unlimited for resources and locks. This can be greater than the initial allocation value (UNLIMITED for infinite limit).

Table 7–5 Values for the RESOURCE_NAME Column

Resource Name	Corresponds to
DML_LOCKS	See "DML_LOCKS" on page 1-61
ENQUEUE_LOCKS	This value is computed by the Oracle Database. See V$ENQUEUE_LOCK on page 6-80 to obtain more information about the enqueue locks.
GES_LOCKS	Global Enqueue Service locks
GES_PROCS	Global Enqueue Service processes
GES_RESS	Global Enqueue Service resources
MAX_SHARED_SERVERS	See "MAX_SHARED_SERVERS" on page 1-96
PARALLEL_MAX_SERVERS	See "PARALLEL_MAX_SERVERS" on page 1-119
PROCESSES	See "PROCESSES" on page 1-128
SESSIONS	See "SESSIONS" on page 1-145
SORT_SEGMENT_LOCKS	This value is computed by the Oracle Database
TEMPORARY_LOCKS	This value is computed by the Oracle Database
TRANSACTIONS	See "TRANSACTIONS" on page 1-162

V$RESTORE_POINT

V$RESTORE_POINT displays information about restore points.

Column	Datatype	Description
SCN	NUMBER	Database SCN when the restore point was created
DATABASE_INCARNATION#	NUMBER	Database incarnation number when the restore point was created
GUARANTEE_FLASHBACK_ DATABASE	VARCHAR2(3)	Indicates whether flashback log files will be kept to ensure a flashback to this point (YES) or not (NO)
STORAGE_SIZE	NUMBER	Approximate number of bytes of disk space currently tied up supporting this restore point. This will only be nonzero for guaranteed restore points.
TIME	TIMESTAMP(9)	Wall-clock time when the restore point was created
RESTORE_POINT_TIME	TIMESTAMP(9)	Time that was specified when the restore point was created. If a time was not specified, this value is NULL.
PRESERVED	VARCHAR2(3)	Indicates whether the restore point must be explicitly deleted (YES) or not (NO)
NAME	VARCHAR2(128)	Name of the restore point

V$RESULT_CACHE_DEPENDENCY

V$RESULT_CACHE_DEPENDENCY displays the depends-on relationship between cached results and dependencies.

Column	Datatype	Description
RESULT_ID	NUMBER	Cached result
DEPEND_ID	NUMBER	Dependency object
OBJECT_NO	NUMBER	Dictionary object number of the dependency object

V$RESULT_CACHE_MEMORY

V$RESULT_CACHE_MEMORY displays all the memory blocks and their status.

Column	Datatype	Description
ID	NUMBER	Unique block identifier (that is, the block number)
CHUNK	NUMBER	Chunk to which the block belongs (the upper 27 bits of the ID)
OFFSET	NUMBER	Offset of the block within its chunk (the lower 5 bits of the ID)
FREE	VARCHAR2(3)	Indicates whether the block is free (YES) or not (NO)
OBJECT_ID	NUMBER	Cache object to which the memory block belongs; NULL unless the memory block is allocated to a cache object (FREE = NO)
POSITION	NUMBER	Position of the block in the cached object; NULL unless the memory block is allocated to a cache object (FREE = NO)

V$RESULT_CACHE_OBJECTS

V$RESULT_CACHE_OBJECTS displays all the objects (both cached results and dependencies) and their attributes.

Column	Datatype	Description
ID	NUMBER	Identifier for the cache object (also the ID of the first block)

Column	Datatype	Description
TYPE	VARCHAR2(10)	Type of the cache object: - Result - Dependency
STATUS	VARCHAR2(9)	Status of the object: - New - Result is still under construction - Published - Result is available for use - Bypass - Result will be bypassed from use - Expired - Result has exceeded expiration time - Invalid - Result is no longer available for use
BUCKET_NO	NUMBER	Internal hash bucket for the object
HASH	NUMBER	Hash value for the object
NAME	VARCHAR2(128)	Name (for example, SQL prefix or PL/SQL function name)
NAMESPACE	VARCHAR2(5)	Namespace: - SQL - PLSQL
CREATION_TIMESTAMP	DATE	Time when the object was created
CREATOR_UID	NUMBER	UID that created the object
DEPEND_COUNT	NUMBER	Number of dependencies (TYPE = Result) or dependents (TYPE = Dependency)
BLOCK_COUNT	NUMBER	Total number of blocks in the cached object
SCN	NUMBER	Build SCN (TYPE = Result) or invalidation SCN (TYPE = Dependency)
COLUMN_COUNT	NUMBER	Number of columns in the cached result[1]
PIN_COUNT	NUMBER	Number of active scans on this result[1]
SCAN_COUNT	NUMBER	Total number of scans initiated on the cached result[1]
ROW_COUNT	NUMBER	Total number of rows in the cached result[1]
ROW_SIZE_MAX	NUMBER	Size of the largest row (in bytes)[1]
ROW_SIZE_MIN	NUMBER	Size of the smallest row (in bytes)[1]
ROW_SIZE_AVG	NUMBER	Average size of a row (in bytes)[1]
BUILD_TIME	NUMBER	Amount of time (in hundredths of a second) it took to build the cached result[1]
LRU_NUMBER	NUMBER	LRU list position (the smaller the value, the more recent the usage)[1]
OBJECT_NO	NUMBER	Dictionary object number of the dependency object[2]
INVALIDATIONS	NUMBER	Number of times the object has invalidated its dependents[2]
SPACE_OVERHEAD	NUMBER	Overhead (in bytes) for the result[1]
SPACE_UNUSED	NUMBER	Unused space (in bytes) for the result[1]
CACHE_ID	VARCHAR2(93)	CacheId for the result (object name if it's a dependency)
CACHE_KEY	VARCHAR2(93)	CacheKey for the result (object name if it's a dependency)

[1] These columns are only valid for TYPE = Result; otherwise, they are NULL.

[2] These columns are only valid for TYPE = Dependency; otherwise, they are NULL.

V$RESULT_CACHE_STATISTICS

V$RESULT_CACHE_STATISTICS displays various Result Cache settings and usage statistics.

Column	Datatype	Description
ID	NUMBER	Statistic number
NAME	VARCHAR2(128)	Name of the statistic (see Table 7–6)
VALUE	NUMBER	Value of the statistic

Table 7–6 V$RESULT_CACHE_STATISTICS Statistics

Statistic Name	Description
Block Size (Bytes)	size of each memory block
Block Count Maximum	Maximum number of memory blocks allowed
Block Count Current	Number of memory blocks currently allocated
Result Size Maximum (Blocks)	Maximum number of blocks allowed for a single result
Create Count Success	Number of cache results successfully created
Create Count Failure	Number of cache results that failed to create
Find Count	Number of cached results that were successfully found
Invalidation Count	Total number of invalidations
Delete Count Invalid	Number of invalid cached results deleted
Delete Count Valid	Number of valid cached results deleted

V$RMAN_BACKUP_JOB_DETAILS

V$RMAN_BACKUP_JOB_DETAILS displays details about backup jobs.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Together, with SESSION_KEY and SESSION_STAMP, used to uniquely identify job output from V$RMAN_OUTPUT
SESSION_STAMP	NUMBER	Together, with SESSION_KEY and SESSION_RECID, used to uniquely identify job output from V$RMAN_OUTPUT
COMMAND_ID	VARCHAR2(33)	Either a user-specified SET COMMAND ID or a unique command ID generated by RMAN
START_TIME	DATE	Start time of the first BACKUP command in the job
END_TIME	DATE	End time of the last BACKUP command in the job
INPUT_BYTES	NUMBER	Sum of all input file sizes backed up by this job
OUTPUT_BYTES	NUMBER	Output size of all pieces generated by this job
STATUS_WEIGHT	NUMBER	Used internally by Enterprise Manager
OPTIMIZED_WEIGHT	NUMBER	Used internally by Enterprise Manager
OBJECT_TYPE_WEIGHT	NUMBER	Used internally by Enterprise Manager
OUTPUT_DEVICE_TYPE	VARCHAR2(17)	Can be DISK, SBT, or *. An * indicates more than one device (in most cases, it will be DISK or SBT).
AUTOBACKUP_COUNT	NUMBER	Number of autobackups performed by this job
AUTOBACKUP_DONE	VARCHAR2(3)	YES or NO, depending upon whether or not a control file autobackup was done as part of this backup job

Column	Datatype	Description
STATUS	VARCHAR2(23)	One of the following values: ■ RUNNING WITH WARNINGS ■ RUNNING WITH ERRORS ■ COMPLETED ■ COMPLETED WITH WARNINGS ■ COMPLETED WITH ERRORS ■ FAILED
INPUT_TYPE	VARCHAR2(13)	Contains one of the following values. If the user command does not satisfy one of them, then preference is given in order, from top to bottom of the list. ■ DB FULL ■ RECVR AREA ■ DB INCR ■ DATAFILE FULL ■ DATAFILE INCR ■ ARCHIVELOG ■ CONTROLFILE ■ SPFILE
OPTIMIZED	VARCHAR2(3)	YES or NO, depending on whether optimization was applied. Applicable to backup jobs only.
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds
COMPRESSION_RATIO	NUMBER	Compression ratio
INPUT_BYTES_PER_SEC	NUMBER	Input read-rate-per-second
OUTPUT_BYTES_PER_SEC	NUMBER	Output write-rate-per-second
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on
INPUT_BYTES_PER_SEC_DISPLAY	VARCHAR2(4000)	Input read-rate-per-second. These values are in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
OUTPUT_BYTES_PER_SEC_DISPLAY	VARCHAR2(4000)	Output write-rate-per-second. These values are in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
TIME_TAKEN_DISPLAY	VARCHAR2(4000)	Time taken, shown in user-displayable format <nn>h:<nn>m:<nn>s

V$RMAN_BACKUP_SUBJOB_DETAILS

V$RMAN_BACKUP_SUBJOB_DETAILS merges similar operations within an RMAN session into a single row. For example, if there are four BACKUP DATAFILE <n> commands, three RECOVERY COPY OF DATAFILE commands, and one BACKUP RECOVERY AREA command, this view will contain three rows - one each for BACKUP, ROLLFORWARD, and COPY_DISK_TO_TAPE operation.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Together with SESSION_KEY and SESSION_STAMP, used to uniquely identify job output from V$RMAN_OUTPUT
SESSION_STAMP	NUMBER	Together with SESSION_KEY and SESSION_RECID, used to uniquely identify job output from V$RMAN_OUTPUT
OPERATION	VARCHAR2(33)	Can be BACKUP, ROLLFORWARD, VALIDATE, or COPY_DISK_TO_TAPE. A row for each suboperation type for the session will be in the output view.

Column	Datatype	Description
COMMAND_ID	VARCHAR2(33)	Either a user-specified SET COMMAND ID or a unique command ID generated by RMAN
START_TIME	DATE	Start time of the first BACKUP command in the job
END_TIME	DATE	End time of the last BACKUP command in the job
INPUT_BYTES	NUMBER	Sum of all input file sizes backed up by this job
OUTPUT_BYTES	NUMBER	Output size of all pieces generated by this job
STATUS_WEIGHT	NUMBER	Used internally by Enterprise Manager
OBJECT_TYPE_WEIGHT	NUMBER	Used internally by Enterprise Manager
OPTIMIZED_WEIGHT	NUMBER	Used internally by Enterprise Manager
OUTPUT_DEVICE_TYPE	VARCHAR2(17)	Can be DISK, SBT, or *. An * indicates more than one device (in most cases, it will be DISK or SBT).
AUTOBACKUP_DONE	VARCHAR2(3)	YES or NO, depending upon whether or not a control file autobackup was done as part of this job
STATUS	VARCHAR2(23)	One of the following values: ■ RUNNING WITH WARNINGS ■ RUNNING WITH ERRORS ■ COMPLETED ■ COMPLETED WITH WARNINGS ■ COMPLETED WITH ERRORS ■ FAILED
INPUT_TYPE	VARCHAR2(13)	Contains one of the following values. If the user command does not satisfy one of them, then preference is given in order, from top to bottom of the list. ■ DB FULL ■ RECVR AREA ■ DB INCR ■ DATAFILE FULL ■ DATAFILE INCR ■ ARCHIVELOG ■ CONTROLFILE ■ SPFILE
OPTIMIZED	VARCHAR2(3)	YES or NO, depending on whether optimization was applied. Applicable to backup jobs only.
AUTOBACKUP_COUNT	NUMBER	Number of autobackups performed by this job
COMPRESSION_RATIO	NUMBER	Compression ratio
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.

V$RMAN_BACKUP_TYPE

V$RMAN_BACKUP_TYPE displays information about RMAN backup types.

Column	Datatype	Description
WEIGHT	NUMBER	Used to set precedence order of different backup types in reports.
INPUT_TYPE	VARCHAR2(13)	Used to represent possible filters used in creating various reporting screens.

V$RMAN_COMPRESSION_ALGORITHM

V$RMAN_COMPRESSION_ALGORITHM provides descriptions of supported compression algorithms. It is used by the RMAN client.

Column	Datatype	Description
ALGORITHM_ID	NUMBER	Algorithm ID
ALGORITHM_NAME	VARCHAR2(64)	Name of the algorithm (for example, BZIP2 or ZLIB)
ALGORITHM_DESCRIPTION	VARCHAR2(64)	Description of the algorithm
ALGORITHM_COMPATIBILITY	VARCHAR2(18)	The required database compatibility level in order for the algorithm to work (for example, 11.0.0 for ZLIB)
DATABASE_COMPATIBILITY	VARCHAR2(18)	Current database compatibility
IS_VALID	VARCHAR2(3)	Whether or not the algorithm is valid with regard to the compatibility setting. Value is YES if ALGORITHM_COMPATIBILITY<=DATABASE_COMPATIBILITY.
IS_DEFAULT	VARCHAR2(3)	Whether or not this is the default algorithm

V$RMAN_CONFIGURATION

V$RMAN_CONFIGURATION lists information about RMAN persistent configuration settings.

Column	Datatype	Description
CONF#	NUMBER	A unique key identifying this configuration record within the target database that owns it.
NAME	VARCHAR2(65)	The type of configuration. All options of the CONFIGURE command are valid types except: ■ CONFIGURE EXCLUDE (described in RC_TABLESPACE) ■ CONFIGURE AUXNAME (described in RC_DATAFILE) ■ CONFIGURE SNAPSHOT CONTROLFILE (stored only in control file)
VALUE	VARCHAR2(1025)	The CONFIGURE command setting. Example: RETENTION POLICY TO RECOVERY WINDOW OF 10 DAYS

V$RMAN_ENCRYPTION_ALGORITHMS

V$RMAN_ENCRYPTION_ALGORITHMS displays supported encryption algorithms. It is used by the RMAN client to validate user-requested algorithms. This view will list AES128, AES192, and AES256 encryption algorithms for the current release. The default algorithm is AES128.

Column	Datatype	Description
ALGORITHM_ID	NUMBER	Number to identify the algorithm
ALGORITHM_NAME	VARCHAR2(64)	Name of the algorithm (for example, AES128, AES192, and AES256)
ALGORITHM_DESCRIPTION	VARCHAR2(64)	Description of the algorithm
IS_DEFAULT	VARCHAR2(3)	YES if this is the default encryption algorithm; otherwise NO.
RESTORE_ONLY	VARCHAR2(3)	NO if this algorithm can be used for backup; otherwise YES.

V$RMAN_OUTPUT

V$RMAN_OUTPUT displays messages reported by RMAN. This is an in-memory view and is not recorded in the controlfile. The view can hold 32768 rows.

Column	Datatype	Description
SID	NUMBER	Session ID of the session which is running this RMAN operation
RECID	NUMBER	Record ID of the corresponding V$RMAN_STATUS row
STAMP	NUMBER	Timestamp of the corresponding V$RMAN_STATUS row
SESSION_RECID	NUMBER	Record ID of the session (corresponding V$RMAN_STATUS row with ROW_LEVEL = 0)
SESSION_STAMP	NUMBER	Timestamp of the session (corresponding V$RMAN_STATUS row with ROW_LEVEL = 0)
OUTPUT	VARCHAR2(129)	Output text reported by RMAN
RMAN_STATUS_RECID	NUMBER	Owning V$RMAN_STATUS record ID
RMAN_STATUS_STAMP	NUMBER	Owning V$RMAN_STATUS record stamp
SESSION_KEY	NUMBER	Session identifier

V$RMAN_STATUS

V$RMAN_STATUS displays the finished and on-going RMAN jobs. For on-going jobs, this view displays progress and status. The jobs which are in progress are stored only in memory while the finished jobs are stored in the controlfile.

Column	Datatype	Description
SID	NUMBER	Session ID of the session which is running this RMAN operation
RECID	NUMBER	Record ID of the row in the controlfile
STAMP	NUMBER	Timestamp of the row (RECID + STAMP is unique)
PARENT_RECID	NUMBER	Record ID of the parent row of this row (corresponding V$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
PARENT_STAMP	NUMBER	Timestamp of the parent row of this row (corresponding V$RMAN_STATUS row with ROW_LEVEL = ROW_LEVEL - 1)
SESSION_RECID	NUMBER	Record ID of the session (corresponding V$RMAN_STATUS row with ROW_LEVEL = 0)
SESSION_STAMP	NUMBER	Timestamp of the session (corresponding V$RMAN_STATUS row with ROW_LEVEL = 0)
ROW_LEVEL	NUMBER	Level of the row. The session has level 0.
ROW_TYPE	VARCHAR2(19)	Type of the row: ■ SESSION ■ COMMAND ■ RECURSIVE OPERATION
COMMAND_ID	VARCHAR2(33)	Command ID set by the RMAN SET COMMAND ID command. If not set, then RMAN will create a unique number.
OPERATION	VARCHAR2(33)	Name of the command in the execution explained by this row

Column	Datatype	Description
STATUS	VARCHAR2(23)	Status of the operation: - RUNNING - RUNNING WITH WARNINGS - RUNNING WITH ERRORS - COMPLETED - COMPLETED WITH WARNINGS - COMPLETED WITH ERRORS - FAILED
MBYTES_PROCESSED	NUMBER	Percentage of the job completed; null if not applicable for the operation
START_TIME	DATE	Start time of the job
END_TIME	DATE	End time of the job
INPUT_BYTES	NUMBER	Number of input bytes read
OUTPUT_BYTES	NUMBER	Number of output bytes written
OPTIMIZED	VARCHAR2(3)	YES, if backup optimization was applied during the backup job. Otherwise, NO.
OBJECT_TYPE	VARCHAR2(13)	Identifies types of objects backed up
OUTPUT_DEVICE_TYPE	VARCHAR2(17)	DISK, SBT_TAPE, or *. An * indicates that output was written to more than one device type.

V$ROLLNAME

V$ROLLNAME lists the names of all online rollback segments. It can only be accessed when the database is open.

Column	Datatype	NULL	Description
USN	NUMBER		Rollback (undo) segment number
NAME	VARCHAR2(30)	NOT NULL	Rollback segment name

V$ROLLSTAT

V$ROLLSTAT contains rollback segment statistics.

Column	Datatype	Description
USN	NUMBER	Rollback segment number
LATCH	NUMBER	Latch for the rollback segment
EXTENTS	NUMBER	Number of extents in the rollback segment
RSSIZE	NUMBER	Size (in bytes) of the rollback segment. This value differs by the number of bytes in one database block from the value of the BYTES column of the *_SEGMENTS view. **See Also:** *Oracle Database Administrator's Guide*
WRITES	NUMBER	Number of bytes written to the rollback segment
XACTS	NUMBER	Number of active transactions
GETS	NUMBER	Number of header gets
WAITS	NUMBER	Number of header waits
OPTSIZE	NUMBER	Optimal size of the rollback segment
HWMSIZE	NUMBER	High-watermark of the rollback segment size
SHRINKS	NUMBER	Number of times the size of a rollback segment decreases

Column	Datatype	Description
WRAPS	NUMBER	Number of times rollback segment is wrapped
EXTENDS	NUMBER	Number of times rollback segment size is extended
AVESHRINK	NUMBER	Average shrink size
AVEACTIVE	NUMBER	Current size of active extents, averaged over time.
STATUS	VARCHAR2(15)	Rollback segment status: ■ ONLINE ■ PENDING OFFLINE ■ OFFLINE ■ FULL
CUREXT	NUMBER	Current extent
CURBLK	NUMBER	Current block

V$ROWCACHE

V$ROWCACHE displays statistics for data dictionary activity. Each row contains statistics for one data dictionary cache.

Column	Datatype	Description
CACHE#	NUMBER	Row cache ID number
TYPE	VARCHAR2(11)	Parent or subordinate row cache type
SUBORDINATE#	NUMBER	Subordinate set number
PARAMETER	VARCHAR2(32)	Name of the initialization parameter that determines the number of entries in the data dictionary cache
COUNT	NUMBER	Total number of entries in the cache
USAGE	NUMBER	Number of cache entries that contain valid data
FIXED	NUMBER	Number of fixed entries in the cache
GETS	NUMBER	Total number of requests for information on the data object
GETMISSES	NUMBER	Number of data requests resulting in cache misses
SCANS	NUMBER	Number of scan requests
SCANMISSES	NUMBER	Number of times a scan failed to find the data in the cache
SCANCOMPLETES	NUMBER	For a list of subordinate entries, the number of times the list was scanned completely
MODIFICATIONS	NUMBER	Number of inserts, updates, and deletions
FLUSHES	NUMBER	Number of times flushed to disk
DLM_REQUESTS	NUMBER	Number of DLM requests
DLM_CONFLICTS	NUMBER	Number of DLM conflicts
DLM_RELEASES	NUMBER	Number of DLM releases

V$ROWCACHE_PARENT

V$ROWCACHE_PARENT displays information for parent objects in the data dictionary. There is one row per lock owner, and one waiter for each object. This row shows the mode held or requested. For objects with no owners or waiters, a single row is displayed.

Column	Datatype	Description
INDX	NUMBER	Index of the row

Column	Datatype	Description
HASH	NUMBER	Hash value
ADDRESS	RAW(4 | 8)	Address of the parent object
CACHE#	NUMBER	Parent cache ID
CACHE_NAME	VARCHAR2(64)	Parent cache name
EXISTENT	VARCHAR2(1)	Indicates whether the object is an existing object
LOCK_MODE	NUMBER	Mode the lock is held in
LOCK_REQUEST	NUMBER	Mode the lock is requested in
TXN	RAW(4)	Transaction currently locking the object
SADDR	RAW(4 | 8)	Address of the session
INST_LOCK_REQUEST	NUMBER	Mode in which instance lock is being requested. This column is only relevant for Real Application Clusters.
INST_LOCK_RELEASE	NUMBER	Whether the instance lock needs to be released. This column is only relevant for Real Application Clusters.
INST_LOCK_TYPE	VARCHAR2(2)	Type of instance lock. This column is only relevant for Real Application Clusters.
INST_LOCK_ID1	RAW(4)	ID associated with the instance lock. This column is only relevant for Real Application Clusters.
INST_LOCK_ID2	RAW(4)	ID associated with the instance lock. This column is only relevant for Real Application Clusters.
KEY	RAW(100)	Contents of the key. This column is only relevant for Real Application Clusters.

V$ROWCACHE_SUBORDINATE

V$ROWCACHE_SUBORDINATE displays information for subordinate objects in the data dictionary.

Column	Datatype	Description
INDX	NUMBER	The index
HASH	NUMBER	The hash value
ADDRESS	RAW(4 | 8)	Address of the subordinate object
CACHE#	NUMBER	The parent cache ID
SUBCACHE#	NUMBER	The subcache ID
SUBCACHE_NAME	VARCHAR2(64)	The subcache name
EXISTENT	VARCHAR2(1)	Whether the object is an existing object
PARENT	RAW(4 | 8)	Address of the parent object
KEY	RAW(100)	The contents of the key

V$RSRC_CONS_GROUP_HISTORY

V$RSRC_CONS_GROUP_HISTORY displays a history of consumer group statistics for each entry in V$RSRC_PLAN_HISTORY that has a non-NULL plan.

Column	Datatype	Description
SEQUENCE#	NUMBER	A sequential counter that uniquely describes the V$RSRC_PLAN_HISTORY entry to which these consumer group statistics apply. When the instance is restarted, this value is reset to zero.

Column	Datatype	Description
ID	NUMBER	Consumer group object ID (a unique number, consistent across database shutdowns and startups)
NAME	VARCHAR2(30)	Name of the consumer group
REQUESTS	NUMBER	Cumulative number of requests that were executed in the consumer group
CPU_WAIT_TIME	NUMBER	Cumulative amount of time that sessions waited for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CPU_WAITS	NUMBER	Cumulative number of times all sessions in the consumer group had to wait for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in the consumer group
YIELDS	NUMBER	Cumulative number of times that sessions in the consumer group had to yield CPU to other sessions because of quantum expiration
ACTIVE_SESS_LIMIT_HIT	NUMBER	Number of times that sessions in the consumer group were queued because the consumer group reached its active session limit
UNDO_LIMIT_HIT	NUMBER	Number of times that queries in the consumer group were cancelled because the consumer group reached its UNDO_POOL limit
SWITCHES_IN_CPU_TIME	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_OUT_CPU_TIME	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_IN_IO_MEGABYTES	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_OUT_IO_MEGABYTES	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_IN_IO_REQUESTS	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SWITCHES_OUT_IO_REQUESTS	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SQL_CANCELED	NUMBER	Number of times that SQL queries running in the consumer group were aborted because they exceeded the Resource Manager plan's SWITCH_TIME limit and CANCEL_SQL was specified as the Resource Manager plan's SWITCH_GROUP
ACTIVE_SESS_KILLED	NUMBER	Number of times that sessions running in the consumer group were terminated because they exceeded the Resource Manager plan's SWITCH_TIME limit and KILL_SESSION was specified as the Resource Manager plan's SWITCH_GROUP
IDLE_SESS_KILLED	NUMBER	Number of times that sessions in the consumer group were killed because they were idle for too long (reached MAX_IDLE_TIME)
IDLE_BLKR_SESS_KILLED	NUMBER	Number of times that sessions in the consumer group were killed because they were idle too long (reached MAX_IDLE_BLOCKER_TIME) and were blocking other sessions
QUEUED_TIME	NUMBER	Total amount of time that sessions in the consumer group have spent in the QUEUED state because of the active session limit
QUEUE_TIME_OUTS	NUMBER	Number of times that requests from sessions in the consumer group timed out because they were queued for too long (reached QUEUEING_P1)
IO_SERVICE_TIME	NUMBER	Cumulative I/O wait time (in microseconds)
IO_SERVICE_WAITS	NUMBER	Total number of wait requests
SMALL_READ_MEGABYTES	NUMBER	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	Number of multiblock megabytes written

Column	Datatype	Description
SMALL_READ_REQUESTS	NUMBER	Number of single block read requests
SMALL_WRITE_REQUESTS	NUMBER	Number of single block write requests
LARGE_READ_REQUESTS	NUMBER	Number of multiblock read requests
LARGE_WRITE_REQUESTS	NUMBER	Number of multiblock write requests

V$RSRC_CONSUMER_GROUP

V$RSRC_CONSUMER_GROUP displays data related to currently active resource consumer groups.

Column	Datatype	Description
ID	NUMBER	Consumer group object ID (a unique number, consistent across database shutdowns and startups)
NAME	VARCHAR2(32)	Name of the consumer group
ACTIVE_SESSIONS	NUMBER	Number of currently active sessions in the consumer group
EXECUTION_WAITERS	NUMBER	Number of currently active sessions waiting for an execution time slice in which they will be able to use CPU
REQUESTS	NUMBER	Cumulative number of requests that were executed in the consumer group
CPU_WAIT_TIME	NUMBER	Cumulative amount of time that sessions waited for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CPU_WAITS	NUMBER	Cumulative number of times all sessions in the consumer group had to wait for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in the consumer group
YIELDS	NUMBER	Cumulative number of times that sessions in the consumer group had to yield CPU to other sessions because of quantum expiration
QUEUE_LENGTH	NUMBER	Number of sessions waiting in the queue
CURRENT_UNDO_CONSUMPTION	NUMBER	Current amount (in KB) of undo consumed by the consumer group
ACTIVE_SESSION_LIMIT_HIT	NUMBER	Number of times that sessions in the consumer group were queued because the consumer group reached its active session limit
UNDO_LIMIT_HIT	NUMBER	Number of times that queries in the consumer group were cancelled because the consumer group reached its UNDO_POOL limit
SWITCHES_IN_CPU_TIME	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_OUT_CPU_TIME	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_TIME limit
SWITCHES_IN_IO_MEGABYTES	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_OUT_IO_MEGABYTES	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_MEGABYTES limit
SWITCHES_IN_IO_REQUESTS	NUMBER	Number of switches into the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SWITCHES_OUT_IO_REQUESTS	NUMBER	Number of switches out of the consumer group because of the Resource Manager plan's SWITCH_IO_REQS limit
SQL_CANCELED	NUMBER	Number of times that SQL queries running in the consumer group were aborted because they exceeded the Resource Manager plan's SWITCH_TIME limit and CANCEL_SQL was specified as the Resource Manager plan's SWITCH_GROUP

Column	Datatype	Description
ACTIVE_SESSIONS_KILLED	NUMBER	Number of times that sessions running in the consumer group were terminated because they exceeded the Resource Manager plan's SWITCH_TIME limit and KILL_SESSION was specified as the Resource Manager plan's SWITCH_GROUP
IDLE_SESSIONS_KILLED	NUMBER	Number of times that sessions in the consumer group were killed because they were idle for too long (reached MAX_IDLE_TIME)
IDLE_BLKR_SESSIONS_KILLED	NUMBER	Number of times that sessions in the consumer group were killed because they were idle too long (reached MAX_IDLE_BLOCKER_TIME) and were blocking other sessions
QUEUED_TIME	NUMBER	Total amount of time that sessions in the consumer group have spent in the QUEUED state because of the active session limit
QUEUE_TIME_OUTS	NUMBER	Number of times that requests from sessions in the consumer group timed out because they were queued for too long (reached QUEUEING_P1)
IO_SERVICE_TIME	NUMBER	Cumulative I/O wait time (in microseconds)
IO_SERVICE_WAITS	NUMBER	Total number of wait requests
SMALL_READ_MEGABYTES	NUMBER	Number of single block megabytes read
SMALL_WRITE_MEGABYTES	NUMBER	Number of single block megabytes written
LARGE_READ_MEGABYTES	NUMBER	Number of multiblock megabytes read
LARGE_WRITE_MEGABYTES	NUMBER	Number of multiblock megabytes written
SMALL_READ_REQUESTS	NUMBER	Number of single block read requests
SMALL_WRITE_REQUESTS	NUMBER	Number of single block write requests
LARGE_READ_REQUESTS	NUMBER	Number of multiblock read requests
LARGE_WRITE_REQUESTS	NUMBER	Number of multiblock write requests

See Also:

- *Oracle Database Administrator's Guide* for information on resource groups
- *Oracle Database PL/SQL Packages and Types Reference* for information on creating resource groups with the DBMS_RESOURCE_MANAGER package

V$RSRC_CONSUMER_GROUP_CPU_MTH

V$RSRC_CONSUMER_GROUP_CPU_MTH displays all resource allocation methods defined for resource consumer groups.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the CPU resource allocation method

See Also:

- *Oracle Database Administrator's Guide* for information on resource allocation methods
- *Oracle Database PL/SQL Packages and Types Reference* on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package
- "V$RSRC_PLAN_CPU_MTH" on page 7-88 for a listing of all resource allocation methods defined for resource plans

V$RSRC_PLAN

V$RSRC_PLAN displays the names of all currently active resource plans.

Column	Datatype	Description
ID	NUMBER	Resource plan ID (a unique number, consistent across database shutdowns and startups). This is also the data dictionary object ID.
NAME	VARCHAR2(32)	Name of the resource plan
IS_TOP_PLAN	VARCHAR2(5)	Indicates whether the resource plan is the current top plan (TRUE) or whether the resource plan is a subplan of the current top plan (FALSE)
CPU_MANAGED	VARCHAR2(3)	Indicates whether the resource plan has parameters that specify a policy for how the Resource Manager should schedule sessions to manage CPU usage (ON) or whether Resource Manager is not managing CPU usage (OFF)

See Also:

- "DBA_RSRC_PLANS" on page 5-27 for a listing of all plans in the database
- *Oracle Database Administrator's Guide* for information on resource plans
- *Oracle Database PL/SQL Packages and Types Reference* for information on defining resource allocation methods for consumer groups with the DBMS_RESOURCE_MANAGER package

V$RSRC_PLAN_CPU_MTH

V$RSRC_PLAN_CPU_MTH displays all available CPU resource allocation methods defined for resource plans.

Column	Datatype	Description
NAME	VARCHAR2(40)	Name of the resource allocation method

See Also:

- "V$RSRC_CONSUMER_GROUP_CPU_MTH" on page 7-87 for a listing of resource allocation methods defined for consumer groups
- *Oracle Database Administrator's Guide* for information on resource plans
- *Oracle Database PL/SQL Packages and Types Reference* for information on defining resource allocation methods for consumer plans with the DBMS_RESOURCE_MANAGER package

V$RSRC_PLAN_HISTORY

V$RSRC_PLAN_HISTORY displays a history of when a resource plan was enabled, disabled, or modified on the instance. Up to 15 of the most recent entries are shown. Once the database is opened, this view shows at least one row. The row with the most recent START_TIME and with END_TIME equal to NULL gives information about the current resource plan.

Column	Datatype	Description
SEQUENCE#	NUMBER	A sequential counter that uniquely describes a row. When the instance is restarted, this value is reset to zero.
ID	NUMBER	Resource plan ID; NULL if the Resource Manager was disabled
NAME	VARCHAR2(30)	Resource plan name; NULL if the Resource Manager was disabled
START_TIME	DATE	Time that the resource plan was enabled
END_TIME	DATE	Time that the resource plan was disabled; NULL if the row contains the current resource plan information
ENABLED_BY_SCHEDULER	VARCHAR2(5)	Indicates whether the plan was enabled by a Job Scheduler window (TRUE) or not (FALSE)
WINDOW_NAME	VARCHAR2(30)	Job Scheduler window that triggered the resource plan event; NULL if a user triggered the resource plan event
ALLOWED_AUTOMATED_ SWITCHES	VARCHAR2(5)	Indicates whether automated plan switches were allowed after this resource plan event (TRUE) or whether automated plan switches were disabled after this resource plan event (FALSE) **See Also:** the SWITCH_PLAN procedure
CPU_MANAGED	VARCHAR2(3)	Indicates whether the resource plan has parameters that specify a policy for how the Resource Manager should schedule sessions to manage CPU usage (ON) or whether Resource Manager is not managing CPU usage (OFF)

V$RSRC_SESSION_INFO

V$RSRC_SESSION_INFO displays Resource Manager statistics per session.

Column	Datatype	Description
SID	NUMBER	Session identifier
CURRENT_CONSUMER_ GROUP_ID	NUMBER	Object ID of the consumer group in which the session currently belongs; NULL if the session has not yet logged in
ORIG_CONSUMER_GROUP_ ID	NUMBER	Object ID of the consumer group in which the session was placed by the consumer group mappings; NULL if the session has not yet logged in This group may not be the current group because the SWITCH_GROUP directive in the current plan may have changed the session's current group. This group may not equal the MAPPED_CONSUMER_GROUP because the MAPPED_CONSUMER_GROUP may not be part of the current plan.
MAPPING_ATTRIBUTE	VARCHAR2(32)	Session attribute that was used to map the session into its original consumer group; NULL if no mapping was used **See Also:** "DBA_RSRC_GROUP_MAPPINGS" on page 5-24 for more details
MAPPED_CONSUMER_GROUP	VARCHAR2(32)	Consumer group to which the session was originally mapped; NULL if no mapping was used This may not correspond to the original consumer group, because the mapped group may not be part of the current plan. **See Also:** "DBA_RSRC_GROUP_MAPPINGS" on page 5-24 for more details

Column	Datatype	Description
STATE	VARCHAR2(32)	Current state of the session: ■ NOT MANAGED - Session has not logged in or the current Resource Manager plan does not require the session to be managed at this point in time ■ RUNNING - Session is currently running on the CPU ■ WAITING FOR CPU - Session is ready to run. It is waiting for a CPU quantum to run. ■ QUEUED - Session is queued because the active session limit was reached ■ IDLE - Session is idle ■ IDLE BLKR - Session is idle and blocking another session ■ WAITING - Session is currently in a wait **See Also:** "V$SESSION_WAIT" on page 8-18 for the wait type ■ WAITING_FOR_IO - Session is waiting to submit an I/O request ■ UNBOUND - Session is not bound to any process ■ EXITING - Session is about to terminate
ACTIVE	VARCHAR2(5)	Indicates whether the session is currently active (TRUE) or not (FALSE). This includes when one of the following conditions is true: ■ Session is in the top call ■ Session has a transaction in progress ■ Session is using temporary space objects ■ Session holds user enqueues
CURRENT_IDLE_TIME	NUMBER	Number of seconds the session has been idle (in states IDLE or IDLE BLKR) while in this consumer group; NULL if the current Resource Manager plan does not require updating this statistic. This value is reset to zero when the session becomes active.
CURRENT_CPU_WAIT_TIME	NUMBER	Amount of time the session has waited for CPU because of resource management (in state WAIT FOR CPU) while in the current consumer group; NULL if the current Resource Manager plan does not require updating this statistic. This does not include waits due to latch or enqueue contention, I/O waits, and so on. If SWITCH_TIME_IN_CALL is used, then this will be reset at the end of every call.
CPU_WAIT_TIME	NUMBER	Cumulative amount of time the session has waited for CPU (in its lifetime) because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CURRENT_CPU_WAITS	NUMBER	Number of times the session had to wait for CPU because of resource management while in this consumer group; NULL if the current Resource Manager plan does not require updating this statistic. This does not include waits due to latch or enqueue contention, I/O waits, and so on. If SWITCH_TIME_IN_CALL is used, then this will be reset at the end of every call.
CPU_WAITS	NUMBER	Cumulative number of times the session had to wait for CPU (in its lifetime) because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.
CURRENT_CONSUMED_CPU_TIME	NUMBER	Amount of CPU time consumed by the session while in the current consumer group; NULL if the current Resource Manager plan does not require updating this statistic. If SWITCH_TIME_IN_CALL is used, then this will be reset at the end of every call.
CONSUMED_CPU_TIME	NUMBER	Cumulative amount of CPU time consumed by the session (in its lifetime)
CURRENT_ACTIVE_TIME	NUMBER	Amount of time the session has been active while in the current consumer group; NULL if the current Resource Manager plan does not require updating this statistic. Active time is time spent running and waiting while executing a call. It does not include the time a session waited for CPU resources. If SWITCH_TIME_IN_CALL is used, then this is reset at the end of every call.
ACTIVE_TIME	NUMBER	Cumulative amount of active time consumed by the session (in its lifetime)

Column	Datatype	Description
CURRENT_QUEUED_TIME	NUMBER	Amount of time the current request from the session has been queued (in state QUEUED). If the session does not have a request currently queued up, then this number will be zero.
QUEUED_TIME	NUMBER	Total amount of time the session has spent in the QUEUED state (in its lifetime)
CURRENT_YIELDS	NUMBER	Number of times the session had to yield the CPU to other sessions (due to quantum expiration) while in the current consumer group; NULL if the current Resource Manager plan does not require updating this statistic. If SWITCH_TIME_IN_CALL is used, then this is reset at the end of every call.
YIELDS	NUMBER	Cumulative number of times the session had to yield CPU to other sessions due to quantum expiration (in its lifetime)
CURRENT_UNDO_ CONSUMPTION	NUMBER	Current amount (in KB) of undo consumed by the session; NULL if the current Resource Manager plan does not have an UNDO_POOL directive
MAX_UNDO_CONSUMPTION	NUMBER	Maximum amount of undo consumed (in KB) during the session's lifetime. This value may not be updated because the current Resource Manager plan may not have an UNDO_POOL directive.
SQL_CANCELED	NUMBER	Number of times SQL queries running in the session were canceled due to exceeding the Resource Manager plan's SWITCH_TIME limit.
QUEUE_TIME_OUTS	NUMBER	Number of times requests from the session timed out because they queued longer than the Resource Manager plan's limit
ESTIMATED_EXECUTION_ LIMIT_HIT	NUMBER	Number of times requests from the session were not run because the optimizer's estimated time to execute the query exceeded the MAX_EST_EXEC_TIME limit
CURRENT_IO_SERVICE_ TIME	NUMBER	Current I/O wait time of the session (in milliseconds)
IO_SERVICE_TIME	NUMBER	Cumulative amount of I/O wait time by the session (in its lifetime)
CURRENT_IO_SERVICE_ WAITS	NUMBER	Current I/O waits by session
IO_SERVICE_WAITS	NUMBER	Cumulative I/O waits by session (in its lifetime)
CURRENT_SMALL_READ_ MEGABYTES	NUMBER	Number of single block megabytes read by the session
SMALL_READ_MEGABYTES	NUMBER	Total number of single block megabytes read by the session (in its lifetime)
CURRENT_LARGE_READ_ MEGABYTES	NUMBER	Number of multiblock megabytes read by the session
LARGE_READ_MEGABYTES	NUMBER	Total number of multiblock megabytes read by the session (in its lifetime)
CURRENT_SMALL_WRITE_ MEGABYTES	NUMBER	Number of single block megabytes written by the session
SMALL_WRITE_MEGABYTES	NUMBER	Total number of single block megabytes written by the session (in its lifetime)
CURRENT_LARGE_WRITE_ MEGABYTES	NUMBER	Number of multiblock megabytes written by the session
LARGE_WRITE_MEGABYTES	NUMBER	Total number of multiblock megabytes written by the session (in its lifetime)
CURRENT_SMALL_READ_ REQUESTS	NUMBER	Number of single block read requests by the session
SMALL_READ_REQUESTS	NUMBER	Total number of single block read requests by the session (in its lifetime)
CURRENT_SMALL_WRITE_ REQUESTS	NUMBER	Number of single block write requests by the session
SMALL_WRITE_REQUESTS	NUMBER	Total number of single block write requests by the session (in its lifetime)
CURRENT_LARGE_READ_ REQUESTS	NUMBER	Number of multiblock read requests by the session
LARGE_READ_REQUESTS	NUMBER	Total number of multiblock read requests by the session (in its lifetime)

Column	Datatype	Description
CURRENT_LARGE_WRITE_REQUESTS	NUMBER	Number of multiblock write requests by the session
LARGE_WRITE_REQUESTS	NUMBER	Total number of multiblock write requests by the session (in its lifetime)

V$RSRCMGRMETRIC

V$RSRCMGRMETRIC displays information about resources consumed and wait times per consumer group.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time for the metric value
END_TIME	DATE	End time for the metric value
INTSIZE_CSEC	NUMBER	Size of the time period
SEQUENCE#	NUMBER	A sequential counter that uniquely describes the V$RSRC_PLAN_HISTORY entry to which these consumer group statistics apply. When the instance is restarted, this value is reset to zero.
CONSUMER_GROUP_ID	NUMBER	Consumer group object ID (a unique number, consistent across database shutdowns and startups)
CONSUMER_GROUP_NAME	VARCHAR2(30)	Name of the consumer group
CPU_CONSUMED_TIME	NUMBER	Cumulative amount of CPU time consumed by all sessions in the consumer group
CPU_WAIT_TIME	NUMBER	Cumulative amount of time that sessions waited for CPU because of resource management. This does not include waits due to latch or enqueue contention, I/O waits, and so on.?
IO_REQUESTS	NUMBER	I/O requests
IO_MEGABYTES	NUMBER	I/O megabytes

V$RSRCMGRMETRIC_HISTORY

V$RSRCMGRMETRIC_HISTORY displays a history (the last one hour) of resource manager metrics, taken from V$RSRCMGRMETRIC. When a resource plan is set, this history is cleared and restarted. This view provides information about resources consumed and wait times per consumer group.

The columns for V$RSRCMGRMETRIC_HISTORY are the same as those for V$RSRCMGRMETRIC.

See Also: "V$RSRCMGRMETRIC" on page 7-92

V$RULE

V$RULE displays rule statistics. This view has a row for every rule loaded into shared memory.

Column	Datatype	Description
RULE_SET_OBJECT_ID	NUMBER	Rule set object ID
EVALUATION_CONTEXT_OBJECT_ID	NUMBER	Evaluation context object ID
RULE_OWNER	VARCHAR2(64)	Owner of the rule
RULE_NAME	VARCHAR2(64)	Name of the rule
RULE_CONDITION	VARCHAR2(64)	Rule condition

Column	Datatype	Description
TRUE_HITS	NUMBER	Number of times the rule evaluated to TRUE
MAYBE_HITS	NUMBER	Number of times the rule evaluated to MAYBE
SQL_EVALUATIONS	NUMBER	Number of evaluations of the rule that were performed by issuing SQL

V$RULE_SET

V$RULE_SET displays rule set statistics. This view has a row for every rule set loaded into shared memory.

> **Note:** Querying the V$RULE_SET view may have a negative impact on performance if a database has a large library cache.

Column	Datatype	Description
OWNER	VARCHAR2(64)	Owner of the rule set
NAME	VARCHAR2(1000)	Name of the rule set
CPU_TIME	NUMBER	Total CPU time (in hundredths of a second) spent in evaluation of the rule set
ELAPSED_TIME	NUMBER	Total elapsed time (in hundredths of a second) spent in evaluation of the rule set
FIRST_LOAD_TIME	DATE	First time the current cached copy of the rule set was loaded
LAST_LOAD_TIME	DATE	Last time the current cached copy of the rule set was loaded
LAST_LOADING_TIME	NUMBER	Total elapsed time (in hundredths of a second) spent to load the rule set the last time it was loaded
SHARABLE_MEM	NUMBER	Shared memory (in bytes) used by the rule set
RELOADS	NUMBER	Number of times the rule set object was reloaded in shared memory
INVALIDATIONS	NUMBER	Number of times the rule set object was invalidated
EVALUATIONS	NUMBER	Number of evaluations on the rule set
FIRST_HIT_EVALUATIONS	NUMBER	Number of evaluations on the rule set, with stop_on_first_hit set to TRUE
SIMPLE_RULES_ONLY_EVALUATIONS	NUMBER	Number of evaluations on the rule set, with simple_rules_only set to TRUE
SQL_FREE_EVALUATIONS	NUMBER	Number of evaluations on the rule set which did not internally issue SQL to evaluate rules
SQL_EXECUTIONS	NUMBER	Total number of SQL statements executed during evaluation of the rule set
CONDITIONS_PROCESSED	NUMBER	Total number of fast (indexed) conditions processed during evaluation of the rule set
TRUE_RULES	NUMBER	Total number of TRUE rules returned during evaluation of the rule set
MAYBE_RULES	NUMBER	Total number of MAYBE rules returned during evaluation of the rule set
VARIABLE_VALUE_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined functions to retrieve variable values (specified by the variable_value_function field in RE$VARIABLE_TYPE) made during evaluation of the rule set
VARIABLE_METHOD_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined functions to retrieve variable method values (specified by the variable_method_function field in RE$VARIABLE_TYPE) made during evaluation of the rule set
EVALUATION_FUNCTION_CALLS	NUMBER	Total number of calls made to user-defined evaluation functions (specified as the evaluation_function argument to the DBMS_RULE_ADM.CREATE_EVALUATION_CONTEXT procedure) made during evaluation of the rule set

V$RULE_SET_AGGREGATE_STATS

V$RULE_SET_AGGREGATE_STATS displays statistics aggregated over all evaluations on all rule sets. This view has a row for each type of statistic.

Column	Datatype	Description
NAME	VARCHAR2(80)	Name of the statistic: ▪ rule set evaluations (all) - Total number of evaluations on all rule sets ▪ rule set evaluations (first_hit) - Total number of evaluations on rule sets with stop_on_first_hit set to TRUE ▪ rule set evaluations (simple_rules_only) - Total number of evaluations on rule sets with simple_rules_only set to TRUE ▪ rule set evaluations (SQL free) - Total number of evaluations on rule sets which did not internally issue SQL to evaluate rules ▪ rule set evaluation time (CPU) - Total CPU time (in hundredths of a second) spent in evaluations on rule sets ▪ rule set evaluation time (elapsed) - Total elapsed time (in hundredths of a second) spent in evaluations on rule sets ▪ rule set SQL executions - Total number of SQL statements executed during evaluations on rule sets ▪ rule set conditions processed - Total number of fast (indexed) conditions processed during evaluations on rule sets ▪ rule set true rules - Total number of TRUE rules returned during evaluations on rule sets ▪ rule set maybe rules - Total number of MAYBE rules returned during evaluations on rule sets ▪ rule set user function calls (variable value function) - Total number of calls made to user-defined functions to retrieve variable values (specified by the variable_value_function field in RE$VARIABLE_TYPE) made during evaluations on rule sets ▪ rule set user function calls (variable method function) - Total number of calls made to user-defined functions to retrieve variable method values (specified by the variable_method_function field in RE$VARIABLE_TYPE) made during evaluations on rule sets ▪ rule set user function calls (evaluation function) - Total number of calls made to user-defined evaluation functions (specified as the evaluation_function argument to the DBMS_RULE_ADM.CREATE_EVALUATION_CONTEXT procedure) made during evaluations on rule sets
VALUE	NUMBER	Statistic value

8

Dynamic Performance (V$) Views: V$SCHEDULER_RUNNING_JOBS to V$XS_SESSION_ROLE

This chapter contains the dynamic performance views V$SCHEDULER_RUNNING_JOBS to V$XS_SESSION_ROLE.

V$SCHEDULER_RUNNING_JOBS

V$SCHEDULER_RUNNING_JOBS displays information about running Scheduler jobs.

Column	Datatype	Description
SESSION_ID	NUMBER	Identifier of the session running the Scheduler job
SESSION_SERIAL_NUM	NUMBER	Session serial number
JOB_ID	NUMBER	ID of the running Scheduler job
PADDR	RAW(4 | 8)	Process address
OS_PROCESS_ID	VARCHAR2(12)	Operating system process ID
SESSION_STAT_CPU	INTERVAL DAY(2) TO SECOND(3)	CPU statistics for the session

V$SECUREFILE_TIMER

V$SECUREFILE_TIMER displays information about time taken (in microseconds) by functions of SecureFiles. These timer values are collected per session.

Column	Datatype	Description
NAME	VARCHAR2(50)	Name of the function
LAYER_ID	NUMBER	ID of the layer that the function belongs to: ■ 0 - Entire Subtree ■ 1 - Delta Updates ■ 2 - Write gather cache ■ 3 - Deduplication ■ 4 - Compression & Encryption ■ 5 - Inode ■ 6 - Space ■ 7 - Utilities ■ 8 - Row-Column Intersection
OWNTIME	NUMBER	Total time taken by the function

Column	Datatype	Description
MAXTIME	NUMBER	Maximum time taken by a single call
MINTIME	NUMBER	Minimum time taken by a single call
INVOCATIONS	NUMBER	Number of times the function was invoked
LAYER_NAME	VARCHAR2(50)	Name of the layer to which the function belongs

V$SEGMENT_STATISTICS

V$SEGMENT_STATISTICS displays information about segment-level statistics.

Column	Datatype	Description
OWNER	VARCHAR2(30)	Owner of the object
OBJECT_NAME	VARCHAR2(30)	Name of the object
SUBOBJECT_NAME	VARCHAR2(30)	Name of the subobject
TABLESPACE_NAME	VARCHAR2(30)	Name of the table space to which the object belongs
TS#	NUMBER	Tablespace number
OBJ#	NUMBER	Dictionary object number of the object
DATAOBJ#	NUMBER	Data object number of the object
OBJECT_TYPE	VARCHAR2(18)	Type of the object
STATISTIC_NAME	VARCHAR2(64)	Name of the statistic
STATISTIC#	NUMBER	Statistic number
VALUE	NUMBER	Statistic value

V$SEGSTAT

V$SEGSTAT displays information about segment-level statistics.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
OBJ#	NUMBER	Dictionary object number
DATAOBJ#	NUMBER	Data object number
STATISTIC_NAME	VARCHAR2(64)	Name of the statistic
STATISTIC#	NUMBER	Statistic number
VALUE	NUMBER	Statistic value

V$SEGSTAT_NAME

V$SEGSTAT_NAME displays information about segment-level statistics properties.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number
NAME	VARCHAR2(64)	Name of the statistic
SAMPLED	VARCHAR2(3)	Indicates whether the statistic was collected by sampling (YES) or not (NO)

V$SERV_MOD_ACT_STATS

V$SERV_MOD_ACT_STATS displays the same set of performance statistics as V$SERVICE_STATS except for a specific combination of service/module/action names.

When aggregation is enabled for the service name, module, and action name, then this view provides the timing and work done for calls issued for the business transaction.

Column	Datatype	Description
AGGREGATION_TYPE	VARCHAR2(21)	Aggregation statistic type: ■ SERVICE_MODULE - Action value is NULL and the entry is an aggregate for all actions within a given module ■ SERVICE_MODULE_ACTION - Action value is NULL only for an empty action, and the aggregation is on the level of service/module/action
SERVICE_NAME	VARCHAR2(64)	Service name from V$SERVICES
MODULE	VARCHAR2(49)	Module name from DBA_ENABLED_AGGREGATIONS
ACTION	VARCHAR2(33)	Action name from DBA_ENABLED_AGGREGATIONS
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2(64)	Derived statistic name from V$STATNAME and V$SESS_TIME_MODEL
VALUE	NUMBER	Cumulative value (in microseconds)

V$SERVICE_EVENT

V$SERVICE_EVENT displays aggregated wait counts and wait times for each wait statistic.

Column	Datatype	Description
SERVICE_NAME	VARCHAR2(64)	Service name from V$SERVICES
SERVICE_NAME_HASH	NUMBER	Service name hash from V$SERVICES
EVENT	VARCHAR2(64)	Name of the wait event; derived statistic name from V$EVENT_NAME
EVENT_ID	NUMBER	Identifier of the event
TOTAL_WAITS	NUMBER	Total amount of time waited for the event by this service (in hundredths of a second)
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event by this service
TIME_WAITED	NUMBER	Time waited for the event (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event by this service (in hundredths of a second)
MAX_WAIT	NUMBER	Maximum time (in hundredths of a second) waited for the event by this service
TIME_WAITED_MICRO	NUMBER	Total time waited for the event (in microseconds)

V$SERVICE_STATS

V$SERVICE_STATS displays a minimal set of performance statistics. These call rate statistics are used for making run-time routing decisions, for tracking service levels, and for per-instance diagnostics per call rate. The elapsed timing for each call provides a relative value across instances for how well a node is processing SQL calls issued under a service name.

When aggregation is enabled for the Service Name, then this view provides the timing and work done for calls issued for the whole service.

Column	Datatype	Description
SERVICE_NAME_HASH	NUMBER	Service name hash from V$SERVICES
SERVICE_NAME	VARCHAR2(64)	Service name from V$SERVICES
STAT_ID	NUMBER	Statistic identifier
STAT_NAME	VARCHAR2(64)	Derived statistic name from V$STATNAME and V$SESS_TIME_MODEL
VALUE	NUMBER	Cumulative value (in microseconds)

V$SERVICE_WAIT_CLASS

V$SERVICE_WAIT_CLASS displays aggregated wait counts and wait times for each wait statistic. An aggregation of these wait classes is used when thresholds are imported.

Column	Datatype	Description
SERVICE_NAME	VARCHAR2(64)	Service name from V$SERVICES
SERVICE_NAME_HASH	NUMBER	Service name hash from V$SERVICES
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
TOTAL_WAITS	NUMBER	Number of times waits of the class occurred for this client
TIME_WAITED	NUMBER	Amount of time (in hundredths of a second) spent in the class by this session

V$SERVICEMETRIC

V$SERVICEMETRIC displays metric values measured on the most recent time interval period for services executing inside the database. Service metrics are measured in 5-second and 1-minute intervals.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin timestamp for the interval period
END_TIME	DATE	End timestamp for the interval period
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Group ID for the service metric group
SERVICE_NAME_HASH	NUMBER	Service name hash
SERVICE_NAME	VARCHAR2(64)	Service name
CTMHASH	NUMBER	Service create timestamp hash value
ELAPSEDPERCALL	NUMBER	Elapsed time per call (in microseconds). This column is deprecated in favor of the DBTIMEPERCALL column.
CPUPERCALL	NUMBER	CPU time per call (in microseconds)
DBTIMEPERCALL	NUMBER	Elapsed time per call (in microseconds)
CALLSPERSEC	NUMBER	Number of user calls per second
DBTIMEPERSEC	NUMBER	Database time per second
GOODNESS	NUMBER	Indicates how attractive a given instance is with respect to processing the workload that is presented to the service. A lower number is better. This number is internally computed based on the GOAL (LONG or SHORT) that is specified for the particular service.
DELTA	NUMBER	Indicates the predicted increase in the goodness for every additional session that is routed to this instance

Column	Datatype	Description
FLAGS	NUMBER	Flags that can be any of the following values: - 0x01 - Service is BLOCKED from accepting new connections - 0x02 - Service is VIOLATING the set threshold on some metric - 0x04 - Goodness is UNKNOWN

V$SERVICEMETRIC_HISTORY

V$SERVICEMETRIC_HISTORY displays a recent history of the metric values measured in predefined time interval periods for services executing inside the database. Service metrics are measured in 5-second and 1-minute intervals.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin timestamp for the interval period
END_TIME	DATE	End timestamp for the interval period
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Group ID for the service metric group
SERVICE_NAME_HASH	NUMBER	Service name hash
SERVICE_NAME	VARCHAR2(64)	Service name
CTMHASH	NUMBER	Service create timestamp hash value
ELAPSEDPERCALL	NUMBER	Elapsed time per call (in microseconds). This column is deprecated in favor of the DBTIMEPERCALL column.
CPUPERCALL	NUMBER	CPU time per call (in microseconds)
DBTIMEPERCALL	NUMBER	Elapsed time per call (in microseconds)
CALLSPERSEC	NUMBER	Number of user calls per second
DBTIMEPERSEC	NUMBER	Database time per second

V$SERVICES

V$SERVICES displays information about the services in the database.

Column	Datatype	Description
SERVICE_ID	NUMBER	Service ID
NAME	VARCHAR2(64)	Name of the service
NAME_HASH	NUMBER	Service name hash
NETWORK_NAME	VARCHAR2(512)	Network name
CREATION_DATE	DATE	Creation date
CREATION_DATE_HASH	NUMBER	Creation date hash
GOAL	VARCHAR2(12)	Service workload management goal: - NONE - SERVICE_TIME - THROUGHPUT
DTP	VARCHAR2(1)	Indicates whether the service is for DTP or distributed transactions including XA transactions (Y) or not (N)
AQ_HA_NOTIFICATION	VARCHAR2(3)	Indicates whether AQ notifications are sent for HA events (YES) or not (NO)

Column	Datatype	Description
CLB_GOAL	VARCHAR2(5)	Connection load balancing goal used with statistics that are sent to the listeners to determine how new connections are distributed: ▪ LONG ▪ SHORT

V$SES_OPTIMIZER_ENV

V$SES_OPTIMIZER_ENV displays the contents of the optimizer environment used by each session. When a new session is first created, it automatically inherits its optimizer environment from the optimizer environment defined at the instance level by V$SYS_OPTIMIZER_ENV. The value of certain parameters can be dynamically modified by issuing an ALTER SESSION statement.

Column	Datatype	Description
SID	NUMBER	Session identifier. This column can be used to join with V$SESSION on the SID column.
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2(40)	Name of the parameter
SQL_FEATURE	VARCHAR2(64)	Associated feature control ID
ISDEFAULT	VARCHAR2(3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2(25)	Value of the parameter for the session

See Also: "V$SYS_OPTIMIZER_ENV" on page 8-65

V$SESS_IO

V$SESS_IO displays I/O statistics for each user session.

Column	Datatype	Description
SID	NUMBER	Session identifier
BLOCK_GETS	NUMBER	Block gets for this session
CONSISTENT_GETS	NUMBER	Consistent gets for this session
PHYSICAL_READS	NUMBER	Physical reads for this session
BLOCK_CHANGES	NUMBER	Block changes for this session
CONSISTENT_CHANGES	NUMBER	Consistent changes for this session

V$SESS_TIME_MODEL

V$SESS_TIME_MODEL displays the session-accumulated time for various operations. The time reported is the total elapsed or CPU time (in microseconds). Any timed operation will buffer at most 5 seconds of time data. Specifically, this means that if a timed operation (such as SQL execution) takes a long period of time to perform, the data published to this view is at most missing 5 seconds of the time accumulated for the operation.

The time values are 8-byte integers and can therefore hold approximately 580,000 years of time before wrapping. Background process time is not included in a statistic value unless the statistic is specifically for background processes.

Column	Datatype	Description
SID	NUMBER	Session ID (same as in V$SESSION)
STAT_ID	NUMBER	Statistic identifier for the time statistic
STAT_NAME	VARCHAR2(64)	Name of the statistic (see Table 8–1)
VALUE	NUMBER	Amount of time (in microseconds) that the session has spent in this operation

Table 8–1 V$SESS_TIME_MODEL and V$SYS_TIME_MODEL Statistics

Statistic Name	Description
DB Time	Amount of elapsed time (in microseconds) spent performing Database user-level calls. This does not include the time spent on instance background processes such as PMON.
DB CPU	Amount of CPU time (in microseconds) spent on database user-level calls. This does not include the CPU time spent on instance background processes such as PMON.
background CPU time	Amount of CPU time (in microseconds) consumed by database background processes.
sequence load elapsed time	Amount of elapsed time spent getting the next sequence number from the data dictionary. If a sequence is cached, then this is the amount of time spent replenishing the cache when it runs out. No time is charged when a sequence number is found in the cache. For non-cached sequences, some time will be charged for every nextval call.
parse time elapsed	Amount of elapsed time spent parsing SQL statements. It includes both soft and hard parse time.
hard parse elapsed time	Amount of elapsed time spent hard parsing SQL statements.
SQL execute elapsed time	Amount of elapsed time SQL statements are executing. Note that for select statements this also includes the amount of time spent performing fetches of query results.
connection management call elapsed time	Amount of elapsed time spent performing session connect and disconnect calls.
failed parse elapsed time	Amount of time spent performing SQL parses which ultimately fail with some parse error.
hard parse (sharing criteria) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from not being able to share an existing cursor in the SQL cache.
hard parse (bind mismatch) elapsed time	Amount of elapsed time spent performing SQL hard parses when the hard parse resulted from bind type or bind size mismatch with an existing cursor in the SQL cache.
PL/SQL execution elapsed time	Amount of elapsed time spent running the PL/SQL interpreter. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing the Java VM.
PL/SQL compilation elapsed time	Amount of elapsed time spent running the PL/SQL compiler.
inbound PL/SQL rpc elapsed time	Time inbound PL/SQL remote procedure calls have spent executing. It includes all time spent recursively executing SQL and JAVA, and therefore is not easily related to "PL/SQL execution elapsed time".
Java execution elapsed time	Amount of elapsed time spent running the Java VM. This does not include time spent recursively executing/parsing SQL statements or time spent recursively executing PL/SQL.

The relationships between the statistics listed in Table 8–1 form two trees in which all the time reported by a child in the tree is contained within the parent in the tree. The following are the relationship trees; the number is the level in the given tree.

```
1) background elapsed time
      2) background cpu time
1) DB time
    2) DB CPU
    2) connection management call elapsed time
    2) sequence load elapsed time
    2) sql execute elapsed time
    2) parse time elapsed
```

```
       3) hard parse elapsed time
          4) hard parse (sharing criteria) elapsed time
            5) hard parse (bind mismatch) elapsed time
       3) failed parse elapsed time
          4) failed parse (out of shared memory) elapsed time
    2) PL/SQL execution elapsed time
    2) inbound PL/SQL rpc elapsed time
    2) PL/SQL compilation elapsed time
    2) Java execution elapsed time
```

The relationship between a parent and a child in the tree indicates containment only. Please keep the following in mind with regard to the tree:

- Children do not necessarily add up to the parent.
- Children are not necessarily exclusive (that is, it is possible that they overlap).
- The union of children does not necessarily cover the whole of the parent.

V$SESSION

`V$SESSION` displays session information for each current session.

Column	Datatype	Description
`SADDR`	`RAW(4 | 8)`	Session address
`SID`	`NUMBER`	Session identifier
`SERIAL#`	`NUMBER`	Session serial number. Used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends and another session begins with the same session ID.
`AUDSID`	`NUMBER`	Auditing session ID
`PADDR`	`RAW(4 | 8)`	Address of the process that owns the session
`USER#`	`NUMBER`	Oracle user identifier
`USERNAME`	`VARCHAR2(30)`	Oracle username
`COMMAND`	`NUMBER`	Command in progress (last statement parsed); for a list of values, see Table 8–2. These values also appear in the `AUDIT_ACTIONS` table.
`OWNERID`	`NUMBER`	Identifier of the user who owns the migratable session; the column contents are invalid if the value is `2147483644` For operations using Parallel Slaves, interpret this value as a 4-byte value. The low-order 2 bytes represent the session number and the high-order bytes represent the instance ID of the query coordinator.
`TADDR`	`VARCHAR2(8)`	Address of the transaction state object
`LOCKWAIT`	`VARCHAR2(8)`	Address of the lock the session is waiting for; NULL if none
`STATUS`	`VARCHAR2(8)`	Status of the session: ■ `ACTIVE` - Session currently executing SQL ■ `INACTIVE` ■ `KILLED` - Session marked to be killed ■ `CACHED` - Session temporarily cached for use by Oracle*XA ■ `SNIPED` - Session inactive, waiting on the client
`SERVER`	`VARCHAR2(9)`	Server type: ■ `DEDICATED` ■ `SHARED` ■ `PSEUDO` ■ `NONE`

Column	Datatype	Description
SCHEMA#	NUMBER	Schema user identifier
SCHEMANAME	VARCHAR2(30)	Schema user name
OSUSER	VARCHAR2(30)	Operating system client user name
PROCESS	VARCHAR2(24)	Operating system client process ID
MACHINE	VARCHAR2(64)	Operating system machine name
TERMINAL	VARCHAR2(30)	Operating system terminal name
PROGRAM	VARCHAR2(48)	Operating system program name
TYPE	VARCHAR2(10)	Session type
SQL_ADDRESS	RAW(4 | 8)	Used with SQL_HASH_VALUE to identify the SQL statement that is currently being executed
SQL_HASH_VALUE	NUMBER	Used with SQL_ADDRESS to identify the SQL statement that is currently being executed
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement that is currently being executed
SQL_CHILD_NUMBER	NUMBER	Child number of the SQL statement that is currently being executed
SQL_EXEC_START	DATE	Time when the execution of the SQL currently executed by this session started; NULL if SQL_ID is NULL
SQL_EXEC_ID	NUMBER	SQL execution identifier; NULL if SQL_ID is NULL or if the execution of that SQL has not yet started (see V$SQL_MONITOR)
PREV_SQL_ADDR	RAW(4 | 8)	Used with PREV_HASH_VALUE to identify the last SQL statement executed
PREV_HASH_VALUE	NUMBER	Used with SQL_HASH_VALUE to identify the last SQL statement executed
PREV_SQL_ID	VARCHAR2(13)	SQL identifier of the last SQL statement executed
PREV_CHILD_NUMBER	NUMBER	Child number of the last SQL statement executed
PREV_EXEC_START	DATE	SQL execution start of the last executed SQL statement
PREV_EXEC_ID	NUMBER	SQL execution identifier of the last executed SQL statement
PLSQL_ENTRY_OBJECT_ID	NUMBER	Object ID of the top-most PL/SQL subprogram on the stack; NULL if there is no PL/SQL subprogram on the stack
PLSQL_ENTRY_SUBPROGRAM_ID	NUMBER	Subprogram ID of the top-most PL/SQL subprogram on the stack; NULL if there is no PL/SQL subprogram on the stack
PLSQL_OBJECT_ID	NUMBER	Object ID of the currently executing PL/SQL subprogram; NULL if executing SQL
PLSQL_SUBPROGRAM_ID	NUMBER	Subprogram ID of the currently executing PL/SQL object; NULL if executing SQL
MODULE	VARCHAR2(48)	Name of the currently executing module as set by calling the DBMS_APPLICATION_INFO.SET_MODULE procedure
MODULE_HASH	NUMBER	Hash value of the MODULE column
ACTION	VARCHAR2(32)	Name of the currently executing action as set by calling the DBMS_APPLICATION_INFO.SET_ACTION procedure
ACTION_HASH	NUMBER	Hash value of the ACTION column
CLIENT_INFO	VARCHAR2(64)	Information set by the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure
FIXED_TABLE_SEQUENCE	NUMBER	This contains a number that increases every time the session completes a call to the database and there has been an intervening select from a dynamic performance table. This column can be used by performance monitors to monitor statistics in the database. Each time the performance monitor looks at the database, it only needs to look at sessions that are currently active or have a higher value in this column than the highest value that the performance monitor saw the last time. All the other sessions have been idle since the last time the performance monitor looked at the database.
ROW_WAIT_OBJ#	NUMBER	Object ID for the table containing the row specified in ROW_WAIT_ROW#

Column	Datatype	Description
ROW_WAIT_FILE#	NUMBER	Identifier for the datafile containing the row specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_BLOCK#	NUMBER	Identifier for the block containing the row specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_ROW#	NUMBER	Current row being locked. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
LOGON_TIME	DATE	Time of logon
LAST_CALL_ET	NUMBER	If the session STATUS is currently ACTIVE, then the value represents the elapsed time (in seconds) since the session has become active. If the session STATUS is currently INACTIVE, then the value represents the elapsed time (in seconds) since the session has become inactive.
PDML_ENABLED	VARCHAR2(3)	This column has been replaced by the PDML_STATUS column
FAILOVER_TYPE	VARCHAR2(13)	Indicates whether and to what extent transparent application failover (TAF) is enabled for the session: ■ NONE - Failover is disabled for this session ■ SESSION - Client is able to fail over its session following a disconnect ■ SELECT - Client is able to fail over queries in progress as well **See Also:** ■ *Oracle Database Concepts* for more information on TAF ■ *Oracle Database Net Services Administrator's Guide* for information on configuring TAF
FAILOVER_METHOD	VARCHAR2(10)	Indicates the transparent application failover method for the session: ■ NONE - Failover is disabled for this session ■ BASIC - Client itself reconnects following a disconnect ■ PRECONNECT - Backup instance can support all connections from every instance for which it is backed up
FAILED_OVER	VARCHAR2(3)	Indicates whether the session is running in failover mode and failover has occurred (YES) or not (NO)
RESOURCE_CONSUMER_GROUP	VARCHAR2(32)	Name of the session's current resource consumer group
PDML_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DML enabled mode. If DISABLED, PARALLEL DML enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DML.
PDDL_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL DDL enabled mode. If DISABLED, PARALLEL DDL enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL DDL.
PQ_STATUS	VARCHAR2(8)	If ENABLED, the session is in a PARALLEL QUERY enabled mode. If DISABLED, PARALLEL QUERY enabled mode is not supported for the session. If FORCED, the session has been altered to force PARALLEL QUERY.
CURRENT_QUEUE_DURATION	NUMBER	If queued (1), the current amount of time the session has been queued. If not currently queued, the value is 0.
CLIENT_IDENTIFIER	VARCHAR2(64)	Client identifier of the session
BLOCKING_SESSION_STATUS	VARCHAR2(11)	Blocking session status: ■ VALID ■ NO HOLDER ■ GLOBAL ■ NOT IN WAIT ■ UNKNOWN
BLOCKING_INSTANCE	NUMBER	Instance identifier of the blocking session

Column	Datatype	Description
BLOCKING_SESSION	NUMBER	Session identifier of the blocking session
SEQ#	NUMBER	A number that uniquely identifies the current or last wait (incremented for each wait)
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting **See Also:** Appendix C, "Oracle Wait Events"
P1TEXT	VARCHAR2(64)	Description of the first wait event parameter
P1	NUMBER	First wait event parameter (in decimal)
P1RAW	RAW(8)	First wait event parameter (in hexadecimal)[1]
P2TEXT	VARCHAR2(64)	Description of the second wait event parameter
P2	NUMBER	Second wait event parameter (in decimal)
P2RAW	RAW(8)	Second wait event parameter (in hexadecimal)[1]
P3TEXT	VARCHAR2(64)	Description of the third wait event parameter
P3	NUMBER	Third wait event parameter (in decimal)
P3RAW	RAW(8)	Third wait event parameter (in hexadecimal)[1]
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2(64)	Name of the class of the wait event
WAIT_TIME	NUMBER	If the session is currently waiting, then the value is 0. If the session is not in a wait, then the value is as follows: ■ > 0 - Value is the duration of the last wait in hundredths of a second ■ -1 - Duration of the last wait was less than a hundredth of a second ■ -2 - Parameter TIMED_STATISTICS was set to false This column has been deprecated in favor of the columns WAIT_TIME_MICRO and STATE.
SECONDS_IN_WAIT	NUMBER	If the session is currently waiting, then the value is the amount of time waited for the current wait. If the session is not in a wait, then the value is the amount of time since the start of the last wait. This column has been deprecated in favor of the columns WAIT_TIME_MICRO and TIME_SINCE_LAST_WAIT_MICRO.
STATE	VARCHAR2(19)	Wait state: ■ WAITING - Session is currently waiting ■ WAITED UNKNOWN TIME - Duration of the last wait is unknown; this is the value when the parameter TIMED_STATISTICS is set to false ■ WAITED SHORT TIME - Last wait was less than a hundredth of a second ■ WAITED KNOWN TIME - Duration of the last wait is specified in the WAIT_TIME column
WAIT_TIME_MICRO	NUMBER	Amount of time waited (in microseconds). If the session is currently waiting, then the value is the time spent in the current wait. If the session is currently not in a wait, then the value is the amount of time waited in the last wait.
TIME_REMAINING_MICRO	NUMBER	Value is interpreted as follows: ■ > 0 - Amount of time remaining for the current wait (in microseconds) ■ 0 - Current wait has timed out ■ -1 - Session can indefinitely wait in the current wait ■ NULL - Session is not currently waiting

Column	Datatype	Description
TIME_SINCE_LAST_WAIT_MICRO	NUMBER	Time elapsed since the end of the last wait (in microseconds). If the session is currently in a wait, then the value is 0.
SERVICE_NAME	VARCHAR2(64)	Service name of the session
SQL_TRACE	VARCHAR2(8)	Indicates whether SQL tracing is enabled (ENABLED) or disabled (DISABLED)
SQL_TRACE_WAITS	VARCHAR2(5)	Indicates whether wait tracing is enabled (TRUE) or not (FALSE)
SQL_TRACE_BINDS	VARCHAR2(5)	Indicates whether bind tracing is enabled (TRUE) or not (FALSE)
SQL_TRACE_PLAN_STATS	VARCHAR2(10)	Frequency at which row source statistics are dumped in the trace files for each cursor: ▪ never ▪ first_execution ▪ all_executions
SESSION_EDITIONID	NUMBER	Reserved for future use
CREATOR_ADDR	RAW(4 | 8)	Address of the creating process or circuit
CREATOR_SERIAL#	NUMBER	Serial number of the creating process or circuit

[1] The P1RAW, P2RAW, and P3RAW columns display the same values as the P1, P2, and P3 columns, except that the numbers are displayed in hexadecimal.

Table 8–2 COMMAND Column of V$SESSION and Corresponding Commands

Number	Command	Number	Command
1	CREATE TABLE	2	INSERT
3	SELECT	4	CREATE CLUSTER
5	ALTER CLUSTER	6	UPDATE
7	DELETE	8	DROP CLUSTER
9	CREATE INDEX	10	DROP INDEX
11	ALTER INDEX	12	DROP TABLE
13	CREATE SEQUENCE	14	ALTER SEQUENCE
15	ALTER TABLE	16	DROP SEQUENCE
17	GRANT OBJECT	18	REVOKE OBJECT
19	CREATE SYNONYM	20	DROP SYNONYM
21	CREATE VIEW	22	DROP VIEW
23	VALIDATE INDEX	24	CREATE PROCEDURE
25	ALTER PROCEDURE	26	LOCK
27	NO-OP	28	RENAME
29	COMMENT	30	AUDIT OBJECT
31	NOAUDIT OBJECT	32	CREATE DATABASE LINK
33	DROP DATABASE LINK	34	CREATE DATABASE
35	ALTER DATABASE	36	CREATE ROLLBACK SEG
37	ALTER ROLLBACK SEG	38	DROP ROLLBACK SEG
39	CREATE TABLESPACE	40	ALTER TABLESPACE
41	DROP TABLESPACE	42	ALTER SESSION
43	ALTER USER	44	COMMIT
45	ROLLBACK	46	SAVEPOINT
47	PL/SQL EXECUTE	48	SET TRANSACTION
49	ALTER SYSTEM	50	EXPLAIN

Table 8–2 (Cont.) COMMAND Column of V$SESSION and Corresponding Commands

Number	Command	Number	Command
51	CREATE USER	52	CREATE ROLE
53	DROP USER	54	DROP ROLE
55	SET ROLE	56	CREATE SCHEMA
57	CREATE CONTROL FILE	59	CREATE TRIGGER
60	ALTER TRIGGER	61	DROP TRIGGER
62	ANALYZE TABLE	63	ANALYZE INDEX
64	ANALYZE CLUSTER	65	CREATE PROFILE
66	DROP PROFILE	67	ALTER PROFILE
68	DROP PROCEDURE	70	ALTER RESOURCE COST
71	CREATE MATERIALIZED VIEW LOG	72	ALTER MATERIALIZED VIEW LOG
73	DROP MATERIALIZED VIEW LOG	74	CREATE MATERIALIZED VIEW
75	ALTER MATERIALIZED VIEW	76	DROP MATERIALIZED VIEW
77	CREATE TYPE	78	DROP TYPE
79	ALTER ROLE	80	ALTER TYPE
81	CREATE TYPE BODY	82	ALTER TYPE BODY
83	DROP TYPE BODY	84	DROP LIBRARY
85	TRUNCATE TABLE	86	TRUNCATE CLUSTER
91	CREATE FUNCTION	92	ALTER FUNCTION
93	DROP FUNCTION	94	CREATE PACKAGE
95	ALTER PACKAGE	96	DROP PACKAGE
97	CREATE PACKAGE BODY	98	ALTER PACKAGE BODY
99	DROP PACKAGE BODY	100	LOGON
101	LOGOFF	102	LOGOFF BY CLEANUP
103	SESSION REC	104	SYSTEM AUDIT
105	SYSTEM NOAUDIT	106	AUDIT DEFAULT
107	NOAUDIT DEFAULT	108	SYSTEM GRANT
109	SYSTEM REVOKE	110	CREATE PUBLIC SYNONYM
111	DROP PUBLIC SYNONYM	112	CREATE PUBLIC DATABASE LINK
113	DROP PUBLIC DATABASE LINK	114	GRANT ROLE
115	REVOKE ROLE	116	EXECUTE PROCEDURE
117	USER COMMENT	118	ENABLE TRIGGER
119	DISABLE TRIGGER	120	ENABLE ALL TRIGGERS
121	DISABLE ALL TRIGGERS	122	NETWORK ERROR
123	EXECUTE TYPE	157	CREATE DIRECTORY
158	DROP DIRECTORY	159	CREATE LIBRARY
160	CREATE JAVA	161	ALTER JAVA
162	DROP JAVA	163	CREATE OPERATOR
164	CREATE INDEXTYPE	165	DROP INDEXTYPE
167	DROP OPERATOR	168	ASSOCIATE STATISTICS
169	DISASSOCIATE STATISTICS	170	CALL METHOD
171	CREATE SUMMARY	172	ALTER SUMMARY
173	DROP SUMMARY	174	CREATE DIMENSION

Table 8–2 (Cont.) COMMAND Column of V$SESSION and Corresponding Commands

Number	Command	Number	Command
175	ALTER DIMENSION	176	DROP DIMENSION
177	CREATE CONTEXT	178	DROP CONTEXT
179	ALTER OUTLINE	180	CREATE OUTLINE
181	DROP OUTLINE	182	UPDATE INDEXES
183	ALTER OPERATOR		

V$SESSION_CONNECT_INFO

V$SESSION_CONNECT_INFO displays information about network connections for the current session.

Column	Datatype	Description
SID	NUMBER	Session identifier (can be used to join this view with V$SESSION)
AUTHENTICATION_TYPE	VARCHAR2(26)	How the user was authenticated: ■ DATABASE - Username/password authentication ■ OS - Operating system external user authentication ■ NETWORK - Network protocol or ANO authentication ■ PROXY - OCI proxy connection authentication ■ SERVER ■ PASSWORD ■ EXTERNAL ADAPTERS ■ INTERNAL ■ GLOBAL ■ EXTERNAL ■ PASSWORD BASED GLOBAL USER
OSUSER	VARCHAR2(30)	External username for this database user
NETWORK_SERVICE_BANNER	VARCHAR2(4000)	Product banners for each Oracle Net service used for this connection (one row per banner)
CLIENT_CHARSET	VARCHAR2(40)	Client character set as specified in the NLS_LANG environment variable or in the OCIEnvNlsCreate() call; Unknown if the Oracle client is older than release 11.1 or the connection is through the JDBC thin driver
CLIENT_CONNECTION	VARCHAR2(13)	Client server connection flags: ■ Heterogeneous ■ Homogeneous
CLIENT_OCI_LIBRARY	VARCHAR2(27)	OCI client library: ■ Home-based ■ Full Instant Client ■ XE Instant Client ■ Light Weight Instant Client ■ OCI ■ Unknown
CLIENT_VERSION	VARCHAR2(40)	Client library version number
CLIENT_DRIVER	VARCHAR2(9)	Client driver name
CLIENT_LOBATTR	VARCHAR2(23)	Client LOB flags: ■ Client Temp Lob Rfc On ■ Client Temp Lob Rfc Off
CLIENT_REGID	NUMBER	Query cache registration ID sent by the client

V$SESSION_CURSOR_CACHE

V$SESSION_CURSOR_CACHE displays information on cursor usage for the current session.

> **Note:** The V$SESSION_CURSOR_CACHE view is not a measure of the effectiveness of the SESSION_CACHED_CURSORS initialization parameter.

Column	Datatype	Description
MAXIMUM	NUMBER	Maximum number of cursors to cache. Once you hit this number, some cursors will need to be closed in order to open more. The value in this column is derived from the initialization parameter SESSION_CACHED_CURSORS.
COUNT	NUMBER	Current number of cursors (whether they are in use or not)
OPENS	NUMBER	Cumulative total of cursor opens minus one. This is because the cursor that is currently open and being used for this query is not counted in the OPENS statistic.
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times an open cursor was found divided by the number of times a cursor was sought

V$SESSION_EVENT

V$SESSION_EVENT displays information on waits for an event by a session. Note that the TIME_WAITED and AVERAGE_WAIT columns will contain a value of zero on those platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set TIMED_STATISTICS to true in the parameter file. Please remember that doing this will have a small negative effect on system performance.

See Also: "TIMED_STATISTICS" on page 1-160

Column	Datatype	Description
SID	NUMBER	ID of the session
EVENT	VARCHAR2(64)	Name of the wait event **See Also:** Appendix C, "Oracle Wait Events"
TOTAL_WAITS	NUMBER	Total number of waits for the event by the session
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event by the session
TIME_WAITED	NUMBER	Total amount of time waited for the event by the session (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event by the session (in hundredths of a second)
MAX_WAIT	NUMBER	Maximum time waited for the event by the session (in hundredths of a second)
TIME_WAITED_MICRO	NUMBER	Total amount of time waited for the event by the session (in microseconds)
EVENT_ID	NUMBER	Identifier of the wait event
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2(64)	Name of the class of the wait event

V$SESSION_FIX_CONTROL

V$SESSION_FIX_CONTROL displays information about Fix Control (enabled/disabled) for the current session.

Column	Datatype	Description
SESSION_ID	NUMBER	Session identifier (can be used to join this view with V$SESSION)
BUGNO	NUMBER	Bug number (as fix control identifier)
VALUE	NUMBER	Current value set for the fix control
SQL_FEATURE	VARCHAR2(64)	Feature control ID
DESCRIPTION	VARCHAR2(64)	Description of the fix control
OPTIMIZER_FEATURE_ENABLE	VARCHAR2(25)	Version on (and after) which the fix is enabled by default
EVENT	NUMBER	Event formerly used to control the fix
IS_DEFAULT	NUMBER	Indicates whether the current value is the same as the default (1) or not (0)

V$SESSION_LONGOPS

V$SESSION_LONGOPS displays the status of various operations that run for longer than 6 seconds (in absolute time). These operations currently include many backup and recovery functions, statistics gathering, and query execution, and more operations are added for every Oracle release.

To monitor query execution progress, you must be using the cost-based optimizer and you must:

- Set the TIMED_STATISTICS or SQL_TRACE parameters to true
- Gather statistics for your objects with the ANALYZE statement or the DBMS_STATS package

You can add information to this view about application-specific long-running operations by using the DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS procedure.

Column	Datatype	Description
SID	NUMBER	Identifier of the session processing the long-running operation. If multiple sessions are cooperating in the long-running operation, then SID corresponds to the main or master session.
SERIAL#	NUMBER	Serial number of the session processing the long-running operation. If multiple sessions are cooperating in the long-running operation, then SERIAL# corresponds to the main or master session. SERIAL# is used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends and another session begins with the same session ID.
OPNAME	VARCHAR2(64)	Brief description of the operation
TARGET	VARCHAR2(64)	Object on which the operation is carried out
TARGET_DESC	VARCHAR2(32)	Description of the target
SOFAR	NUMBER	Units of work done so far
TOTALWORK	NUMBER	Total units of work
UNITS	VARCHAR2(32)	Units of measurement
START_TIME	DATE	Starting time of the operation
LAST_UPDATE_TIME	DATE	Time when statistics were last updated for the operation
TIMESTAMP	DATE	Timestamp specific to the operation

Column	Datatype	Description
TIME_REMAINING	NUMBER	Estimate (in seconds) of time remaining for the operation to complete
ELAPSED_SECONDS	NUMBER	Number of elapsed seconds from the start of the operations
CONTEXT	NUMBER	Context
MESSAGE	VARCHAR2(512)	Statistics summary message
USERNAME	VARCHAR2(30)	User ID of the user performing the operation
SQL_ADDRESS	RAW(4 | 8)	Used with the value of the SQL_HASH_VALUE column to identify the SQL statement associated with the operation
SQL_HASH_VALUE	NUMBER	Used with the value of the SQL_ADDRESS column to identify the SQL statement associated with the operation
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement associated with the long operation, if any
SQL_PLAN_HASH_VALUE	NUMBER	SQL plan hash value; NULL if SQL_ID is NULL
SQL_EXEC_START	DATE	Time when the execution of the SQL started; NULL if SQL_ID is NULL
SQL_EXEC_ID	NUMBER	SQL execution identifier (see V$SQL_MONITOR)
SQL_PLAN_LINE_ID	NUMBER	SQL plan line ID corresponding to the long operation; NULL if the long operation is not associated with a line of the execution plan
SQL_PLAN_OPERATION	VARCHAR2(30)	Plan operation name; NULL if SQL_PLAN_LINE_ID is NULL
SQL_PLAN_OPTIONS	VARCHAR2(30)	Plan operation options; NULL if SQL_PLAN_LINE_ID is NULL
QCSID	NUMBER	Session identifier of the parallel coordinator

See Also: *Oracle Database PL/SQL Packages and Types Reference* for more information on DBMS_APPLICATION_INFO.SET_SESSION_LONGOPS

V$SESSION_OBJECT_CACHE

V$SESSION_OBJECT_CACHE displays object cache statistics for the current user session on the local server (instance).

Column	Datatype	Description
PINS	NUMBER	Number of object pins or look-ups in the cache
HITS	NUMBER	Number of object pins that found the object already in the cache
TRUE_HITS	NUMBER	Number of object pins that found the object already in the cache and in the desired state (thus, not requiring refresh from the database)
HIT_RATIO	NUMBER	Ratio of HITS / PINS
TRUE_HIT_RATIO	NUMBER	Ratio of TRUE_HITS/PINS
OBJECT_REFRESHES	NUMBER	Number of objects in the cache that were refreshed with a new value from the database
CACHE_REFRESHES	NUMBER	Number of times the whole cache (all objects) were refreshed
OBJECT_FLUSHES	NUMBER	Number of objects in the cache that were flushed to the database
CACHE_FLUSHES	NUMBER	Number of times the whole cache (all objects) were flushed to the database
CACHE_SHRINKS	NUMBER	Number of times the cache was shrunk to the optimal size
CACHED_OBJECTS	NUMBER	Number of objects currently cached
PINNED_OBJECTS	NUMBER	Number of objects currently pinned
CACHE_SIZE	NUMBER	Current size of the cache (in bytes)
OPTIMAL_SIZE	NUMBER	Optimal size of the cache (in bytes)
MAXIMUM_SIZE	NUMBER	Maximum size of the cache (in bytes)

V$SESSION_WAIT

V$SESSION_WAIT displays the current or last wait for each session.

Column	Datatype	Description
SID	NUMBER	Session identifier; maps to V$SESSION.SID
SEQ#	NUMBER	A number that uniquely identifies the current or last wait (incremented for each wait)
WAIT_ID	NUMBER	Wait identifier
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting **See Also:** Appendix C, "Oracle Wait Events"
P1TEXT	VARCHAR2(64)	Description of the first wait event parameter
P1	NUMBER	First wait event parameter (in decimal)
P1RAW	RAW(8)	First wait event parameter (in hexadecimal)[1]
P2TEXT	VARCHAR2(64)	Description of the second wait event parameter
P2	NUMBER	Second wait event parameter (in decimal)
P2RAW	RAW(8)	Second wait event parameter (in hexadecimal)[1]
P3TEXT	VARCHAR2(64)	Description of the third wait event parameter
P3	NUMBER	Third wait event parameter (in decimal)
P3RAW	RAW(8)	Third wait event parameter (in hexadecimal)[1]
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2(64)	Name of the class of the wait event
WAIT_TIME	NUMBER	If the session is currently waiting, then the value is 0. If the session is not in a wait, then the value is as follows: ■ > 0 - Value is the duration of the last wait in hundredths of a second ■ -1 - Duration of the last wait was less than a hundredth of a second ■ -2 - Parameter TIMED_STATISTICS was set to false This column has been deprecated in favor of the columns WAIT_TIME_MICRO and STATE.
SECONDS_IN_WAIT	NUMBER	If the session is currently waiting, then the value is the amount of time waited for the current wait. If the session is not in a wait, then the value is the amount of time since the start of the last wait. This column has been deprecated in favor of the columns WAIT_TIME_MICRO and TIME_SINCE_LAST_WAIT_MICRO.
STATE	VARCHAR2(19)	Wait state: ■ WAITING - Session is currently waiting ■ WAITED UNKNOWN TIME - Duration of the last wait is unknown; this is the value when the parameter TIMED_STATISTICS is set to false ■ WAITED SHORT TIME - Last wait was less than a hundredth of a second ■ WAITED KNOWN TIME - Duration of the last wait is specified in the WAIT_TIME column
WAIT_TIME_MICRO	NUMBER	Amount of time waited (in microseconds). If the session is currently waiting, then the value is the time spent in the current wait. If the session is currently not in a wait, then the value is the amount of time waited in the last wait.

Column	Datatype	Description
TIME_REMAINING_MICRO	NUMBER	Value is interpreted as follows: ▪ > 0 - Amount of time remaining for the current wait (in microseconds) ▪ 0 - Current wait has timed out ▪ -1 - Session can indefinitely wait in the current wait ▪ NULL - Session is not currently waiting
TIME_SINCE_LAST_WAIT_MICRO	NUMBER	Time elapsed since the end of the last wait (in microseconds). If the session is currently in a wait, then the value is 0.

[1] The P1RAW, P2RAW, and P3RAW columns display the same values as the P1, P2, and P3 columns, except that the numbers are displayed in hexadecimal.

See Also: "TIMED_STATISTICS" on page 1-160 and Appendix C, "Oracle Wait Events"

V$SESSION_WAIT_CLASS

V$SESSION_WAIT_CLASS displays the time spent in various wait event operations on a per-session basis.

Column	Datatype	Description
SID	NUMBER	Session identifier
SERIAL#	NUMBER	Serial number
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
TOTAL_WAITS	NUMBER	Number of times waits of the class occurred for the session
TIME_WAITED	NUMBER	Amount of time spent in the wait class by the session

V$SESSION_WAIT_HISTORY

V$SESSION_WAIT_HISTORY displays the last 10 wait events for each active session.

Column	Datatype	Description
SID	NUMBER	Session identifier
SEQ#	NUMBER	Sequence of wait events; 1 is the most recent
EVENT#	NUMBER	Event number
EVENT	VARCHAR2(64)	Resource or event for which the session is waiting
P1TEXT	VARCHAR2(64)	Description of the first wait event parameter
P1	NUMBER	First wait event parameter (in decimal)
P2TEXT	VARCHAR2(64)	Description of the second wait event parameter
P2	NUMBER	Second wait event parameter (in decimal)
P3TEXT	VARCHAR2(64)	Description of the third wait event parameter
P3	NUMBER	Third wait event parameter (in decimal)
WAIT_TIME	NUMBER	Amount of time waited (in hundredths of a second)
WAIT_TIME_MICRO	NUMBER	Amount of time waited (in microseconds)
TIME_SINCE_LAST_WAIT_MICRO	NUMBER	Time elapsed (in microseconds) since the end of the previous wait in the wait history

V$SESSMETRIC

V$SESSMETRIC displays the metric values for all sessions.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
SESSION_ID	NUMBER	Session ID
SESSION_SERIAL_NUM	NUMBER	Session serial number
CPU	NUMBER	CPU usage
PHYSICAL_READS	NUMBER	Number of physical reads
LOGICAL_READS	NUMBER	Number of logical reads
PGA_MEMORY	NUMBER	PGA size at the end of the interval
HARD_PARSES	NUMBER	Number of hard parses
SOFT_PARSES	NUMBER	Number of soft parses
PHYSICAL_READ_PCT	NUMBER	Physical read ratio
LOGICAL_READ_PCT	NUMBER	Logical read ratio

V$SESSTAT

V$SESSTAT displays user session statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V$STATNAME view.

Column	Datatype	Description
SID	NUMBER	Session identifier
STATISTIC#	NUMBER	Statistic number **Note:** Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
VALUE	NUMBER	Statistic value

See Also: "V$STATNAME" on page 8-55 and Appendix E, "Statistics Descriptions"

V$SGA

V$SGA displays summary information about the system global area (SGA).

Column	Datatype	Description
NAME	VARCHAR2(20)	SGA component group
VALUE	NUMBER	Memory size (in bytes)

V$SGA_CURRENT_RESIZE_OPS

V$SGA_CURRENT_RESIZE_OPS displays information about SGA resize operations which are currently in progress. An operation can be a grow or a shrink of a dynamic SGA component. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Desired value of the parameter after the resize
CURRENT_SIZE	NUMBER	Current value of the parameter
START_TIME	DATE	Start time of the operation
LAST_UPDATE_TIME	DATE	Last time progress was made for the operation

V$SGA_DYNAMIC_COMPONENTS

V$SGA_DYNAMIC_COMPONENTS displays information about the dynamic SGA components. This view summarizes information based on all completed SGA resize operations since instance startup. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
CURRENT_SIZE	NUMBER	Current size of the component
MIN_SIZE	NUMBER	Minimum size of the component since instance startup
MAX_SIZE	NUMBER	Maximum size of the component since instance startup
USER_SPECIFIED_SIZE	NUMBER	Value of the user parameter for the component
OPER_COUNT	NUMBER	Number of operations since instance startup
LAST_OPER_TYPE	VARCHAR2(13)	Last completed operation for the component: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL
LAST_OPER_MODE	VARCHAR2(9)	Mode of the last completed operation: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
LAST_OPER_TIME	DATE	Start time of the last completed operation
GRANULE_SIZE	NUMBER	Granularity of the grow or the shrink operation

V$SGA_DYNAMIC_FREE_MEMORY

V$SGA_DYNAMIC_FREE_MEMORY displays information about the amount of SGA memory available for future dynamic SGA resize operations.

Column	Datatype	Description
CURRENT_SIZE	NUMBER	Amount of available memory (in bytes)

V$SGA_RESIZE_OPS

V$SGA_RESIZE_OPS displays information about the last 800 completed SGA resize operations. This does not include in-progress operations. All sizes are expressed in bytes.

Column	Datatype	Description
COMPONENT	VARCHAR2(64)	Component name
OPER_TYPE	VARCHAR2(13)	Operation type: ■ STATIC ■ INITIALIZING ■ DISABLED ■ GROW ■ SHRINK ■ SHRINK_CANCEL
OPER_MODE	VARCHAR2(9)	Operation mode: ■ MANUAL ■ DEFERRED ■ IMMEDIATE
PARAMETER	VARCHAR2(80)	Name of the parameter for the resize operation
INITIAL_SIZE	NUMBER	Parameter value at the start of the operation
TARGET_SIZE	NUMBER	Requested value of the parameter after the resize
FINAL_SIZE	NUMBER	Real value of the parameter after the resize
STATUS	VARCHAR2(9)	Completion status of the operation: ■ INACTIVE ■ PENDING ■ COMPLETE ■ CANCELLED ■ ERROR
START_TIME	DATE	Start time of the operation
END_TIME	DATE	End time of the operation

V$SGA_TARGET_ADVICE

V$SGA_TARGET_ADVICE displays information about the SGA_TARGET initialization parameter.

Column	Datatype	Description
SGA_SIZE	NUMBER	Size of the SGA
SGA_SIZE_FACTOR	NUMBER	Ratio between the SGA_SIZE and the current size of the SGA
ESTD_DB_TIME	NUMBER	Estimated DB_TIME for this SGA_SIZE

Column	Datatype	Description
ESTD_DB_TIME_FACTOR	NUMBER	Ratio between ESTD_DB_TIME and DB_TIME for the current size of the SGA
ESTD_PHYSICAL_READS	NUMBER	Estimated number of physical reads

V$SGAINFO

V$SGAINFO displays size information about the SGA, including the sizes of different SGA components, the granule size, and free memory.

Column	Datatype	Description
NAME	VARCHAR2(32)	Name of the SGA statistic
BYTES	NUMBER	Size of the statistic (in bytes)
RESIZEABLE	VARCHAR2(3)	Indicates whether the statistic is resizeable (YES) or not (NO)

V$SGASTAT

V$SGASTAT displays detailed information on the system global area (SGA).

Column	Datatype	Description
POOL	VARCHAR2(12)	Designates the pool in which the memory in NAME resides: ■ shared pool - Memory is allocated from the shared pool ■ large pool - Memory is allocated from the large pool ■ java pool - Memory is allocated from the Java pool ■ streams pool - Memory is allocated from the Streams pool
NAME	VARCHAR2(26)	SGA component name
BYTES	NUMBER	Memory size (in bytes)

V$SHARED_POOL_ADVICE

V$SHARED_POOL_ADVICE displays information about estimated parse time in the shared pool for different pool sizes. The sizes range from 10% of the current shared pool size or the amount of pinned library cache memory (whichever is higher) to 200% of the current shared pool size, in equal intervals. The value of the interval depends on the current size of the shared pool.

Column	Datatype	Description
SHARED_POOL_SIZE_FOR_ESTIMATE	NUMBER	Shared pool size for the estimate (in megabytes)
SHARED_POOL_SIZE_FACTOR	NUMBER	Size factor with respect to the current shared pool size
ESTD_LC_SIZE	NUMBER	Estimated memory in use by the library cache (in megabytes)
ESTD_LC_MEMORY_OBJECTS	NUMBER	Estimated number of library cache memory objects in the shared pool of the specified size
ESTD_LC_TIME_SAVED	NUMBER	Estimated elapsed parse time saved (in seconds), owing to library cache memory objects being found in a shared pool of the specified size. This is the time that would have been spent in reloading the required objects in the shared pool had they been aged out due to insufficient amount of available free memory.
ESTD_LC_TIME_SAVED_FACTOR	NUMBER	Estimated parse time saved factor with respect to the current shared pool size

Column	Datatype	Description
ESTD_LC_LOAD_TIME	NUMBER	Estimated elapsed time (in seconds) for parsing in a shared pool of the specified size
ESTD_LC_LOAD_TIME_FACTOR	NUMBER	Estimated load time factor with respect to the current shared pool size
ESTD_LC_MEMORY_OBJECT_HITS	NUMBER	Estimated number of times a library cache memory object was found in a shared pool of the specified size

V$SHARED_POOL_RESERVED

V$SHARED_POOL_RESERVED displays statistics that help you tune the reserved pool and space within the shared pool.

Column	Datatype	Description
FREE_SPACE	NUMBER	Total amount of free space on the reserved list[1]
AVG_FREE_SIZE	NUMBER	Average size of the free memory on the reserved list[1]
FREE_COUNT	NUMBER	Number of free pieces of memory on the reserved list[1]
MAX_FREE_SIZE	NUMBER	Size of the largest free piece of memory on the reserved list[1]
USED_SPACE	NUMBER	Total amount of used memory on the reserved list[1]
AVG_USED_SIZE	NUMBER	Average size of the used memory on the reserved list[1]
USED_COUNT	NUMBER	Number of used pieces of memory on the reserved list[1]
MAX_USED_SIZE	NUMBER	Size of the largest used piece of memory on the reserved list[1]
REQUESTS	NUMBER	Number of times that the reserved list was searched for a free piece of memory[1]
REQUEST_MISSES	NUMBER	Number of times the reserved list did not have a free piece of memory to satisfy the request, and started flushing objects from the LRU list[1]
LAST_MISS_SIZE	NUMBER	Request size of the last request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list[1]
MAX_MISS_SIZE	NUMBER	Request size of the largest request miss, when the reserved list did not have a free piece of memory to satisfy the request and started flushing objects from the LRU list[1]
REQUEST_FAILURES	NUMBER	Number of times that no memory was found to satisfy a request (that is, the number of times the error ORA-04031 occurred)[2]
LAST_FAILURE_SIZE	NUMBER	Request size of the last failed request (that is, the request size for the last ORA-04031 error)[2]
ABORTED_REQUEST_THRESHOLD	NUMBER	Minimum size of a request which signals an ORA-04031 error without flushing objects[2]
ABORTED_REQUESTS	NUMBER	Number of requests that signalled an ORA-04031 error without flushing objects[2]
LAST_ABORTED_SIZE	NUMBER	Last size of the request that returned an ORA-04031 error without flushing objects from the LRU list[2]

[1] These columns are valid only if the initialization parameter SHARED_POOL_RESERVED_SIZE is set to a valid value.

[2] These columns contain values which are valid even if SHARED_POOL_RESERVED_SIZE is not set.

V$SHARED_SERVER

V$SHARED_SERVER displays information on the shared server processes.

Column	Datatype	Description
NAME	VARCHAR2(4)	Name of the server
PADDR	RAW(4 | 8)	Server's process address
STATUS	VARCHAR2(16)	Server status: EXEC - Executing SQL WAIT (ENQ) - Waiting for a lock WAIT (SEND) - Waiting to send data to user WAIT (COMMON) - Idle; waiting for a user request WAIT (RESET) - Waiting for a circuit to reset after a break QUIT - Terminating
MESSAGES	NUMBER	Number of messages processed
BYTES	NUMBER	Total number of bytes in all messages
BREAKS	NUMBER	Number of breaks
CIRCUIT	RAW(4 | 8)	Address of the circuit currently being serviced
IDLE	NUMBER	Total idle time (in hundredths of a second)
BUSY	NUMBER	Total busy time (in hundredths of a second)
REQUESTS	NUMBER	Total number of requests taken from the common queue in this server's lifetime

V$SHARED_SERVER_MONITOR

V$SHARED_SERVER_MONITOR displays information for tuning the shared server.

Column	Datatype	Description
MAXIMUM_CONNECTIONS	NUMBER	Highest number of virtual circuits in use at one time since the instance started. If this value reaches the value set for the CIRCUITS initialization parameter, then consider raising the value of CIRCUITS. **See Also:** "CIRCUITS" on page 1-28
MAXIMUM_SESSIONS	NUMBER	Highest number of shared server sessions in use at one time since the instance started. If this reaches the value set for the SHARED_SERVER_SESSIONS initialization parameter, then consider raising the value of SHARED_SERVER_SESSIONS. **See Also:** "SHARED_SERVER_SESSIONS" on page 1-149
SERVERS_STARTED	NUMBER	Total number of shared servers started since the instance started (but not including those started during startup)
SERVERS_TERMINATED	NUMBER	Total number of shared servers stopped by Oracle since the instance started
SERVERS_HIGHWATER	NUMBER	Highest number of servers running at one time since the instance started. If this value reaches the value set for the MAX_SHARED_SERVERS initialization parameter, then consider raising the value of SHARED_SERVERS. **See Also:** "SHARED_SERVERS" on page 1-150

V$SORT_SEGMENT

V$SORT_SEGMENT displays information about every sort segment in a given instance. The view is only updated when the tablespace is of the TEMPORARY type.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(31)	Name of the tablespace
SEGMENT_FILE	NUMBER	File number of the first extent

Column	Datatype	Description
SEGMENT_BLOCK	NUMBER	Block number of the first extent
EXTENT_SIZE	NUMBER	Extent size
CURRENT_USERS	NUMBER	Number of active users of the segment
TOTAL_EXTENTS	NUMBER	Total number of extents in the segment
TOTAL_BLOCKS	NUMBER	Total number of blocks in the segment
USED_EXTENTS	NUMBER	Extents allocated to active sorts
USED_BLOCKS	NUMBER	Blocks allocated to active sorts
FREE_EXTENTS	NUMBER	Extents not allocated to any sort
FREE_BLOCKS	NUMBER	Blocks not allocated to any sort
ADDED_EXTENTS	NUMBER	Number of extent allocations
EXTENT_HITS	NUMBER	Number of times an unused extent was found in the pool
FREED_EXTENTS	NUMBER	Number of deallocated extents
FREE_REQUESTS	NUMBER	Number of requests to deallocate
MAX_SIZE	NUMBER	Maximum number of extents ever used
MAX_BLOCKS	NUMBER	Maximum number of blocks ever used
MAX_USED_SIZE	NUMBER	Maximum number of extents used by all sorts
MAX_USED_BLOCKS	NUMBER	Maximum number of blocks used by all sorts
MAX_SORT_SIZE	NUMBER	Maximum number of extents used by an individual sort
MAX_SORT_BLOCKS	NUMBER	Maximum number of blocks used by an individual sort
RELATIVE_FNO	NUMBER	Relative file number of the sort segment header

V$SPPARAMETER

V$SPPARAMETER displays information about the contents of the server parameter file. If a server parameter file was not used to start the instance, then each row of the view will contain FALSE in the ISSPECIFIED column.

Column	Datatype	Description
SID	VARCHAR2(80)	SID for which the parameter is defined
NAME	VARCHAR2(80)	Name of the parameter
TYPE	VARCHAR2(11)	Parameter type
VALUE	VARCHAR2(255)	Parameter value (null if a server parameter file was not used to start the instance)
DISPLAY_VALUE	VARCHAR2(255)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISSPECIFIED	VARCHAR2(6)	Indicates whether the parameter was specified in the server parameter file (TRUE) or not (FALSE)
ORDINAL	NUMBER	Position (ordinal number) of the parameter value (0 if a server parameter file was not used to start the instance). Useful only for parameters whose values are lists of strings.
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update (null if a server parameter file was not used to start the instance)

V$SQL

V$SQL lists statistics on shared SQL areas without the GROUP BY clause and contains one row for each child of the original SQL text entered. Statistics displayed in V$SQL are normally updated at the end of query execution. However, for long running queries, they are updated every 5 seconds. This makes it easy to see the impact of long running SQL statements while they are still in progress.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor
SQL_FULLTEXT	CLOB	Full text for the SQL statement exposed as a CLOB column. The full text of a SQL statement can be retrieved using this column instead of joining with the V$SQLTEXT dynamic performance view.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
SHARABLE_MEM	NUMBER	Amount of shared memory used by the child cursor (in bytes)
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of the child cursor (in bytes)
RUNTIME_MEM	NUMBER	Fixed amount of memory required during the execution of the child cursor
SORTS	NUMBER	Number of sorts that were done for the child cursor
LOADED_VERSIONS	NUMBER	Indicates whether the context heap is loaded (1) or not (0)
OPEN_VERSIONS	NUMBER	Indicates whether the child cursor is locked (1) or not (0)
USERS_OPENING	NUMBER	Number of users executing the statement
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Number of executions that took place on this object since it was brought into the library cache
PX_SERVERS_EXECUTIONS	NUMBER	Total number of executions performed by parallel execution servers (0 when the statement has never been executed in parallel)
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
USERS_EXECUTING	NUMBER	Number of users executing the statement
LOADS	NUMBER	Number of times the object was either loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	Timestamp of the parent creation time
INVALIDATIONS	NUMBER	Number of times this child cursor has been invalidated
PARSE_CALLS	NUMBER	Number of parse calls for this child cursor
DISK_READS	NUMBER	Number of disk reads for this child cursor
DIRECT_WRITES	NUMBER	Number of direct writes for this child cursor
BUFFER_GETS	NUMBER	Number of buffer gets for this child cursor
APPLICATION_WAIT_TIME	NUMBER	Application wait time (in microseconds)
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time (in microseconds)
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time (in microseconds)
USER_IO_WAIT_TIME	NUMBER	User I/O Wait Time (in microseconds)
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time (in microseconds)
JAVA_EXEC_TIME	NUMBER	Java execution time (in microseconds)
ROWS_PROCESSED	NUMBER	Total number of rows the parsed SQL statement returns
COMMAND_TYPE	NUMBER	Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement was executed
OPTIMIZER_COST	NUMBER	Cost of this query given by the optimizer
OPTIMIZER_ENV	RAW(2000)	Optimizer environment

Column	Datatype	Description
OPTIMIZER_ENV_HASH_VALUE	NUMBER	Hash value for the optimizer environment
PARSING_USER_ID	NUMBER	User ID of the user who originally built this child cursor
PARSING_SCHEMA_ID	NUMBER	Schema ID that was used to originally build this child cursor
PARSING_SCHEMA_NAME	VARCHAR2(30)	Schema name that was used to originally build this child cursor
KEPT_VERSIONS	NUMBER	Indicates whether this child cursor has been marked to be kept pinned in the cache using the DBMS_SHARED_POOL package
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
TYPE_CHK_HEAP	RAW(4)	Descriptor of the type check heap for this child cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
OLD_HASH_VALUE	NUMBER	Old SQL hash value
PLAN_HASH_VALUE	NUMBER	Numeric representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
CHILD_NUMBER	NUMBER	Number of this child cursor
SERVICE	VARCHAR2(64)	Service name
SERVICE_HASH	NUMBER	Hash value for the name listed in the SERVICEcolumn
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	Hash value of the module listed in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed, which is set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	Hash value of the action listed in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction failed to serialize, producing ORA-08177 errors, per cursor
OUTLINE_CATEGORY	VARCHAR2(64)	If an outline was applied during construction of the cursor, then this column displays the category of that outline. Otherwise the column is left blank.
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing, executing, and fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing, executing, and fetching
OUTLINE_SID	NUMBER	Outline session identifier
CHILD_ADDRESS	RAW(4 | 8)	Address of the child cursor
SQLTYPE	NUMBER	Denotes the version of the SQL language used for this statement
REMOTE	VARCHAR2(1)	Indicates whether the cursor is remote mapped (Y) or not (N)
OBJECT_STATUS	VARCHAR2(19)	Status of the cursor: ■ VALID - Valid, authorized without errors ■ VALID_AUTH_ERROR - Valid, authorized with authorization errors ■ VALID_COMPILE_ERROR - Valid, authorized with compilation errors ■ VALID_UNAUTH - Valid, unauthorized ■ INVALID_UNAUTH - Invalid, unauthorized ■ INVALID - Invalid, unauthorized but keep the timestamp
LITERAL_HASH_VALUE	NUMBER	Hash value of the literals which are replaced with system-generated bind variables and are to be matched, when CURSOR_SHARING is used. This is not the hash value for the SQL statement. If CURSOR_SHARING is not used, then the value is 0.
LAST_LOAD_TIME	VARCHAR2(19)	Time at which the query plan was loaded into the library cache

Column	Datatype	Description
IS_OBSOLETE	VARCHAR2(1)	Indicates whether the cursor has become obsolete (Y) or not (N). This can happen if the number of child cursors is too large.
IS_BIND_SENSITIVE	VARCHAR2(1)	Indicates whether the cursor is bind sensitive (Y) or not (N). A query is considered bind-sensitive if the optimizer peeked at one of its bind variable values when computing predicate selectivities and where a change in a bind variable value may cause the optimizer to generate a different plan.
IS_BIND_AWARE	VARCHAR2(1)	Indicates whether the cursor is bind aware (Y) or not (N). A query is considered bind-aware if it has been marked to use extended cursor sharing. The query would already have been marked as bind-sensitive.
IS_SHAREABLE	VARCHAR2(1)	Indicates whether the cursor can be shared (Y) or not (N)
CHILD_LATCH	NUMBER	Child latch number that is protecting the cursor
SQL_PROFILE	VARCHAR2(64)	SQL profile used for this statement, if any
SQL_PATCH	VARCHAR2(30)	SQL patch used for this statement, if any
SQL_PLAN_BASELINE	VARCHAR2(30)	SQL plan baseline used for this statement, if any
PROGRAM_ID	NUMBER	Program identifier
PROGRAM_LINE#	NUMBER	Program line number
EXACT_MATCHING_SIGNATURE	NUMBER	Signature calculated on the normalized SQL text. The normalization includes the removal of white space and the uppercasing of all non-literal strings.
FORCE_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to FORCE
LAST_ACTIVE_TIME	DATE	TIme at which the query plan was last active
BIND_DATA	RAW(2000)	Bind data
TYPECHECK_MEM	NUMBER	???

V$SQL_BIND_CAPTURE

V$SQL_BIND_CAPTURE displays information on bind variables used by SQL cursors. Each row in the view contains information for one bind variable defined in a cursor. This includes:

- Reference to the cursor defining the bind variable

  (hash_value, address) for the parent cursor and (hash_value, child_address) for the child cursor.

- Bind metadata

  Name, position, datatype, character set ID, precision, scale, and maximum length of the bind variable.

- Bind data

  One of the bind values used for the bind variable during a past execution of its associated SQL statement. Bind values are not always captured for this view. Bind values are displayed by this view only when the type of the bind variable is simple (this excludes LONG, LOB, and ADT datatypes) and when the bind variable is used in the WHERE or HAVING clauses of the SQL statement.

Bind capture is disabled when the STATISTICS_LEVEL initialization parameter is set to BASIC. This view can be joined with V$SQLAREA on (HASH_VALUE, ADDRESS) and with V$SQL on (HASH_VALUE, CHILD_ADDRESS).

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the parent cursor
HASH_VALUE	NUMBER	Hash value of the parent cursor in the library cache. The hash value is a fixed index for the view and should always be used to speed up access to the view.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
CHILD_ADDRESS	RAW(4 | 8)	Address of the parent cursor
CHILD_NUMBER	NUMBER	Child cursor number
NAME	VARCHAR2(30)	Name of the bind variable
POSITION	NUMBER	Position of the bind variable in the SQL statement
DUP_POSITION	NUMBER	If the binding is performed by name and the bind variable is duplicated, then this column gives the position of the primary bind variable.
DATATYPE	NUMBER	Internal identifier for the bind datatype
DATATYPE_STRING	VARCHAR2(15)	Textual representation of the bind datatype
CHARACTER_SID	NUMBER	National character set identifier
PRECISION	NUMBER	Precision (for numeric binds)
SCALE	NUMBER	Scale (for numeric binds)
MAX_LENGTH	NUMBER	Maximum bind length
WAS_CAPTURED	VARCHAR2(3)	Indicates whether the bind value was captured (YES) or not (NO)
LAST_CAPTURED	DATE	Date when the bind value was captured. Bind values are captured when SQL statements are executed. To limit the overhead, binds are captured at most every 15 minutes for a given cursor.
VALUE_STRING	VARCHAR2(4000)	Value of the bind represented as a string
VALUE_ANYDATA	ANYDATA	Value of the bind represented using the ANYDATA datatype. This representation is useful to programmatically decode the value of the bind variable.

V$SQL_BIND_DATA

V$SQL_BIND_DATA describes, for each distinct bind variable in each cursor owned by the session querying this view:

- Actual bind data, if the bind variable is user defined
- The underlying literal, if the CURSOR_SHARING parameter is set to FORCE and the bind variable is system generated. (System-generated binds have a value of 256 in the SHARED_FLAG2 column.)

Column	Datatype	Description
CURSOR_NUM	NUMBER	Cursor number for this bind
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
SHARED_MAX_LEN	NUMBER	Shared maximum length for this bind from the shared cursor object associated with this bind
PRIVATE_MAX_LEN	NUMBER	Private maximum length for this bind sent from the client
ARRAY_SIZE	NUMBER	Maximum number of array elements (for array binds only)
PRECISION	NUMBER	Precision (for numeric binds)
SCALE	NUMBER	Scale (for numeric binds)
SHARED_FLAG	NUMBER	Shared bind data flags
SHARED_FLAG2	NUMBER	Shared bind data flags (continued)

Column	Datatype	Description
BUF_ADDRESS	RAW(4 | 8)	Bind buffer memory address
BUF_LENGTH	NUMBER	Bind buffer length
VAL_LENGTH	NUMBER	Actual bind value length
BUF_FLAG	NUMBER	Bind buffer flags
INDICATOR	NUMBER	Bind indicator
VALUE	VARCHAR2(4000)	Contents of the bind buffer

V$SQL_BIND_METADATA

V$SQL_BIND_METADATA describes, for each distinct bind variable in each cursor owned by the session querying this view:

- Bind metadata provided by the client, if the bind variable is user defined
- Metadata based on the underlying literal, if the CURSOR_SHARING parameter is set to FORCE and the bind variable is system-generated.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Memory address of the child cursor that owns this bind variable
POSITION	NUMBER	Bind position
DATATYPE	NUMBER	Bind datatype
MAX_LENGTH	NUMBER	Maximum length of the bind value
ARRAY_LEN	NUMBER	Maximum number of array elements (for array binds only)
BIND_NAME	VARCHAR2(30)	User-defined or system-generated bind variable name (if used)

V$SQL_CURSOR

V$SQL_CURSOR displays debugging information for each cursor associated with the session querying this view.

Column	Datatype	Description
CURNO	NUMBER	Cursor number
FLAG	NUMBER	Flags set in the cursor
STATUS	VARCHAR2(9)	Status of the cursor; that is, what state the cursor is in
PARENT_HANDLE	RAW(4 | 8)	Pointer to the parent cursor handle
PARENT_LOCK	RAW(4 | 8)	Pointer to the parent cursor lock
CHILD_LOCK	RAW(4 | 8)	Pointer to the child cursor lock
CHILD_PIN	RAW(4 | 8)	Pointer to the child cursor pin
PERS_HEAP_MEM	NUMBER	Total amount of memory allocated from persistent heap for this cursor
WORK_HEAP_MEM	NUMBER	Total amount of memory allocated from the work heap for this cursor
BIND_VARS	NUMBER	Total number of bind positions in the query currently parsed into this cursor
DEFINE_VARS	NUMBER	Total number of define variables in the query currently parsed into this cursor
BIND_MEM_LOC	VARCHAR2(64)	Which memory heap the bind variables are stored in: either the UGA or the CGA
INST_FLAG	VARCHAR2(64)	Instantiation object flags
INST_FLAG2	VARCHAR2(64)	Instantiation object flags (continued)

V$SQL_JOIN_FILTER

V$SQL_JOIN_FILTER displays performance information about the characteristics of join filters when they are used for a parallel cursor. (A join filter is a bitmap filter applied to table rows prior to a join operation in order to avoid parallel communication.)

Column	Datatype	Description
QC_SESSION_ID	NUMBER	QC (Query Coordinator) session ID of the given cursor for the given parallel query
QC_INSTANCE_ID	NUMBER	QC (Query Coordinator) instance ID of the given cursor for the given parallel query
SQL_PLAN_HASH_VALUE	NUMBER	SQL plan hash value of the given cursor for the given parallel query
LENGTH	NUMBER	Total size of the join filter field
BITS_SET	NUMBER	Number of bits set in this filter
FILTERED	NUMBER	Number of rows seen by the join filter
PROBED	NUMBER	Number of rows of the right table that have be tested against the bitmap filter. This is the sum of the filtered rows plus the non-filtered rows.
ACTIVE	NUMBER	Whether the filter is active (Yes) or not (No)

V$SQL_MONITOR

V$SQL_MONITOR displays SQL statements whose execution have been (or are being) monitored by Oracle. An entry is created in V$SQL_MONITOR every time the execution of a SQL statement is being monitored. SQL monitoring is automatically started when a SQL statement runs parallel or when it has consumed at least 5 seconds of CPU or I/O time.

When the SQL statement being monitored is executing, statistics in V$SQL_MONITOR are generally refreshed in near real time, once every second. Once the execution ends, monitoring information is not deleted immediately. Instead, it is kept in V$SQL_MONITOR for at least one minute. The entry will eventually be deleted to reclaim its space as new statements are monitored.

Column	Datatype	Description
KEY	NUMBER	Artificial join key to efficiently join V$SQL_MONITOR with its corresponding plan level monitoring statistics stored in V$SQL_PLAN_MONITOR
STATUS	VARCHAR2(19)	SQL execution status: ■ EXECUTING - SQL statement is still executing ■ DONE (ERROR) - Execution terminated with an error ■ DONE (FIRST N ROWS) - Execution terminated by the application before all rows were fetched ■ DONE (ALL ROWS) - Execution terminated and all rows were fetched ■ DONE - Execution terminated (parallel execution)
FIRST_REFRESH_TIME	DATE	Time when monitoring of the SQL statement started, generally a few seconds after execution start time
LAST_REFRESH_TIME	DATE	Time when statistics in V$SQL_MONITOR were last updated for the SQL statement. Statistics are generally refreshed every second when the statement executes.
REFRESH_COUNT	NUMBER	Number of times V$SQL_MONITOR statistics have been refreshed (generally once every second when the SQL statement executes)

Column	Datatype	Description
SID	NUMBER	Session identifier executing (or having executed) the SQL statement being monitored
PROCESS_NAME	VARCHAR2(5)	Process name identifier executing (or having executed)the statement; ora if the process is foreground, else the background process name (for example, p001 for PX server p001)
SQL_ID	VARCHAR2(13)	SQL identifier of the statement being monitored
SQL_EXEC_START	DATE	Time when the execution started
SQL_EXEC_ID	NUMBER	Execution identifier. Together, the three columns SQL_ID, SQL_EXEC_START, and SQL_EXEC_ID represent the execution key. The execution key is used to uniquely identify one execution of the SQL statement.
SQL_PLAN_HASH_VALUE	NUMBER	SQL Plan hash value
SQL_CHILD_ADDRESS	RAW(4 | 8)	Address of the child cursor (can be used with SQL_ID to join with V$SQL)
SESSION_SERIAL#	NUMBER	Session serial number executing the statement being monitored
PX_SERVER#	NUMBER	Logical parallel execution server process number executing (or having executed) the statement being monitored; NULL if this monitoring entry is not associated with an execution server. This is a logical number within the parallel server set (see SERVER# in V$PX_SESSION).
PX_SERVER_GROUP	NUMBER	Logical parallel execution server group number to which PX_SERVER# belongs (see SERVER_GROUP in V$PX_SESSION); NULL if this monitoring entry is not associated with a parallel execution server. This value is generally 1 unless the SQL statement has one or more parallel sub-queries.
PX_SERVER_SET	NUMBER	Number (1 or 2) of the logical set of parallel execution servers to which PX_SERVER# belongs (see SERVER_SET in V$PX_SESSION); NULL if this monitoring entry is not associated with a parallel execution server
PX_QCINST_ID	NUMBER	Instance identifier where the parallel execution coordinator runs; NULL if PX_SERVER# is NULL
PX_QCSID	NUMBER	Session identifier for the parallel execution coordinator; NULL if PX_SERVER# is NULL
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds); updated as the statement executes
CPU_TIME	NUMBER	CPU time (in microseconds); updated as the statement executes
FETCHES	NUMBER	Number of fetches associated with the SQL statement; updated as the statement executes
BUFFER_GETS	NUMBER	Number of buffer get operations; updated as the statement executes
DISK_READS	NUMBER	Number of disk reads; updated as the statement executes
DIRECT_WRITES	NUMBER	Number of direct writes; updated as the statement executes
APPLICATION_WAIT_TIME	NUMBER	Application wait time (in microseconds); updated as the statement executes
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time (in microseconds); updated as the statement executes
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time (in microseconds); updated as the statement executes
USER_IO_WAIT_TIME	NUMBER	User I/O Wait Time (in microseconds); updated as the statement executes
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time (in microseconds); updated as the statement executes
JAVA_EXEC_TIME	NUMBER	Java execution time (in microseconds); updated as the statement executes

V$SQL_OPTIMIZER_ENV

V$SQL_OPTIMIZER_ENV displays the contents of the optimizer environment used to build the execution plan of a SQL cursor. This view can be joined with V$SQLAREA on (HASH_VALUE, ADDRESS) and with V$SQL on (HASH_VALUE, CHILD_ADDRESS).

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the parent cursor
HASH_VALUE	NUMBER	Hash value of the parent cursor in the library cache. The hash value is the fixed index for the view and should be used when querying V$SQL_OPTIMIZER_ENV to avoid scanning the entire library cache.
SQL_ID	VARCHAR2(13)	SQL identifier
CHILD_ADDRESS	RAW(4 | 8)	Address of the child cursor
CHILD_NUMBER	NUMBER	Child cursor number
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2(40)	Name of the parameter
ISDEFAULT	VARCHAR2(3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2(25)	Value of the parameter

V$SQL_PLAN

V$SQL_PLAN contains the execution plan information for each child cursor loaded in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V$SQLAREA to add the cursor-specific information.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for the cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$SQL to add the child cursor-specific information.
OPERATION	VARCHAR2(30)	Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)	A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(40)	Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER	Object number of the table or the index
OBJECT_OWNER	VARCHAR2(31)	Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)	Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)	Alias for the object
OBJECT_TYPE	VARCHAR2(20)	Type of the object
OPTIMIZER	VARCHAR2(20)	Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
ID	NUMBER	A number assigned to each step in the execution plan
PARENT_ID	NUMBER	ID of the next execution step that operates on the output of the current step

Column	Datatype	Description
DEPTH	NUMBER	Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER	Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER	Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER	Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER	Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER	Estimate, by the cost-based optimizer, of the number of bytes produced by the operation
OTHER_TAG	VARCHAR2(35)	Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)	Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)	Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER	Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)	Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)	Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER	CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER	I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER	Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)	Expressions produced by the operation
TIME	NUMBER	Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)	Name of the query block
REMARKS	VARCHAR2(4000)	Remarks
OTHER_XML	CLOB	Provides extra information specific to an execution step of the execution plan. The content of this column is structured using XML since multiple pieces of information can be stored there. This includes: ▪ Name of the schema against which the query was parsed ▪ Release number of the Oracle Database that produced the explain plan ▪ Hash value associated with the execution plan ▪ Name (if any) of the outline or the SQL profile used to build the execution plan ▪ Indication of whether or not dynamic sampling was used to produce the plan ▪ The outline data, a set of optimizer hints that can be used to regenerate the same plan

V$SQL_PLAN_MONITOR

V$SQL_PLAN_MONITOR displays plan level monitoring statistics for each SQL statement found in V$SQL_MONITOR. Each row in V$SQL_PLAN_MONITOR corresponds to an operation of the execution plan being monitored. As with V$SQL_MONITOR, statistics exposed in V$SQL_PLAN_MONITOR are generally updated every second when the statement executes. These statistics are recycled on the same basis as V$SQL_MONITOR.

To eliminate the overhead of SQL plan monitoring, statistics collected for each operation of the plan don\qt record timing information such as elapsed time, CPU time, or I/O time. Instead, this timing information can be estimated quite accurately by joining V$SQL_PLAN_MONITOR with V$ACTIVE_SESSION_HISTORY on SQL_ID, SQL_EXEC_START, SQL_EXEC_ID, and SQL_PLAN_LINE_ID (simply named PLAN_LINE_ID in V$SQL_PLAN_MONITOR). The result of that join is a sample of the activity performed by each operation in the plan, from which an estimate of CPU time and wait time can be derived. This can be achieved by breaking statement level monitoring time statistics found in V$SQL_MONITOR in proportion to the number of samples found in V$ACTIVE_SESSION_HISTORY for the corresponding activity type.

Column	Datatype	Description
KEY	NUMBER	Foreign key to efficiently join V$SQL_PLAN_MONITOR with V$SQL_MONITOR (see V$SQL_MONITOR)
STATUS	VARCHAR2(19)	SQL execution status: ■ EXECUTING - SQL statement is still executing ■ DONE (ERROR) - Execution terminated with an error ■ DONE (FIRST N ROWS) - Execution terminated by the application before all rows were fetched ■ DONE (ALL ROWS) - Execution terminated and all rows were fetched ■ DONE - Execution terminated (parallel execution)
FIRST_REFRESH_TIME	DATE	Time when monitoring of the SQL statement started
LAST_REFRESH_TIME	DATE	Time when statistics were last updated for the SQL statement
FIRST_CHANGE_TIME	DATE	First time a row was produced by this operation
LAST_CHANGE_TIME	DATE	Last time a row was produced by this operation
REFRESH_COUNT	NUMBER	Number of times statistics have been refreshed
SID	NUMBER	Session identifier executing (or having executed) the SQL statement being monitored
PROCESS_NAME	VARCHAR2(5)	Process name identifier
SQL_ID	VARCHAR2(13)	SQL identifier
SQL_EXEC_START	DATE	Time when the execution started
SQL_EXEC_ID	NUMBER	Execution identifier
SQL_PLAN_HASH_VALUE	NUMBER	SQL plan hash value
SQL_CHILD_ADDRESS	RAW(4 | 8)	Address of the child cursor
PLAN_LINE_ID	NUMBER	Plan line number for the entry
PLAN_OPERATION	VARCHAR2(30)	Plan operation name (from V$SQL_PLAN)
PLAN_OPTIONS	VARCHAR2(30)	Plan option name (from V$SQL_PLAN)
STARTS	NUMBER	Number of times this operation was executed. For example, an operation is executed multiple times when it is on the right side of a nested-loop join (once for each row of the left input of that nested-loop join).

Column	Datatype	Description
OUTPUT_ROWS	NUMBER	Number of rows produced by this operation since the execution started. This number is cumulated for all executions of this operation. Divide by the value of the STARTS column to compute the average number of rows per execution of the operation.
WORKAREA_MEM	NUMBER	Amount of memory (in bytes) used by the operation when the query is executing; NULL if the execution is done. This applies only to operations using a work area, such as sort, hash-join, group-by, and so on.
WORKAREA_MAX_MEM	NUMBER	Maximum value (in bytes) for WORKAREA_MEM; NULL if the operation is not using a work area. When the execution is finished, this value will hold the maximum amount of memory consumed by this operation during the execution of the statement.
WORKAREA_TEMPSEG	NUMBER	Amount of temporary space (in bytes) used by the operation when the query is executing; NULL if the operation has not spilled to disk or if the execution is finished
WORKAREA_MAX_TEMPSEG	NUMBER	Maximum value (in bytes) for WORKAREA_TEMPSEG; NULL if this operation never spilled to disk. When the execution is done, this value will hold the maximum amount of temporary space consumed by this operation during the entire execution.

V$SQL_PLAN_STATISTICS

V$SQL_PLAN_STATISTICS provides execution statistics at the row source level for each child cursor.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V$SQLAREA to locate the parent cursor.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache
PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line)
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$SQL to locate the child cursor using this area.
OPERATION_ID	NUMBER	A number assigned to each step in the execution plan
EXECUTIONS	NUMBER	Number of times this cursor has been executed
LAST_STARTS	NUMBER	Number of times this operation has been started, during the last execution
STARTS	NUMBER	Number of times this operation has been started, accumulated over the past executions
LAST_OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, during the last execution
OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, accumulated over the past executions
LAST_CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, during the last execution. Buffers are usually retrieved in consistent mode for queries.
CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, accumulated over the past executions. Buffers are usually retrieved in consistent mode for queries.
LAST_CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, during the last execution. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, accumulated over the past executions. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.

Column	Datatype	Description
LAST_DISK_READS	NUMBER	Number of physical disk reads performed by the operation, during the last execution
DISK_READS	NUMBER	Number of physical disk reads performed by the operation, accumulated over the past executions
LAST_DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, during the last execution
DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, accumulated over the past executions
LAST_ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, during the last execution
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, accumulated over the past executions

V$SQL_PLAN_STATISTICS_ALL

V$SQL_PLAN_STATISTICS_ALL contains memory usage statistics for row sources that use SQL memory (sort or hash-join). This view concatenates information in V$SQL_PLAN with execution statistics from V$SQL_PLAN_STATISTICS and V$SQL_WORKAREA.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. The two columns ADDRESS and HASH_VALUE can be used to join with V$SQLAREA to add the cursor-specific information.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache
PLAN_HASH_VALUE	NUMBER	Numerical representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line)
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this execution plan. The columns ADDRESS, HASH_VALUE, and CHILD_NUMBER can be used to join with V$SQL to add the child cursor-specific information.
OPERATION	VARCHAR2(30)	Name of the internal operation performed in this step (for example, TABLE ACCESS)
OPTIONS	VARCHAR2(30)	A variation on the operation described in the OPERATION column (for example, FULL)
OBJECT_NODE	VARCHAR2(40)	Name of the database link used to reference the object (a table name or view name). For local queries that use parallel execution, this column describes the order in which output from operations is consumed.
OBJECT#	NUMBER	Object number of the table or the index
OBJECT_OWNER	VARCHAR2(31)	Name of the user who owns the schema containing the table or index
OBJECT_NAME	VARCHAR2(31)	Name of the table or index
OBJECT_ALIAS	VARCHAR2(65)	Alias for the object
OBJECT_TYPE	VARCHAR2(20)	Type of the object
OPTIMIZER	VARCHAR2(20)	Current mode of the optimizer for the first row in the plan (statement line), for example, CHOOSE. When the operation is a database access (for example, TABLE ACCESS), this column indicates whether or not the object is analyzed.
ID	NUMBER	A number assigned to each step in the execution plan
PARENT_ID	NUMBER	ID of the next execution step that operates on the output of the current step

Column	Datatype	Description
DEPTH	NUMBER	Depth (or level) of the operation in the tree. It is not necessary to issue a CONNECT BY statement to get the level information, which is generally used to indent the rows from the PLAN_TABLE table. The root operation (statement) is level 0.
POSITION	NUMBER	Order of processing for all operations that have the same PARENT_ID
SEARCH_COLUMNS	NUMBER	Number of index columns with start and stop keys (that is, the number of columns with matching predicates)
COST	NUMBER	Cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
CARDINALITY	NUMBER	Estimate, by the cost-based optimizer, of the number of rows produced by the operation
BYTES	NUMBER	Estimate, by the cost-based optimizer, of the number of bytes produced by the operation
OTHER_TAG	VARCHAR2(35)	Describes the contents of the OTHER column. See EXPLAIN PLAN for values.
PARTITION_START	VARCHAR2(5)	Start partition of a range of accessed partitions
PARTITION_STOP	VARCHAR2(5)	Stop partition of a range of accessed partitions
PARTITION_ID	NUMBER	Step that computes the pair of values of the PARTITION_START and PARTITION_STOP columns
OTHER	VARCHAR2(4000)	Other information specific to the execution step that users may find useful. See EXPLAIN PLAN for values.
DISTRIBUTION	VARCHAR2(20)	Stores the method used to distribute rows from producer query servers to consumer query servers
CPU_COST	NUMBER	CPU cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
IO_COST	NUMBER	I/O cost of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
TEMP_SPACE	NUMBER	Temporary space usage of the operation (sort or hash-join) as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
ACCESS_PREDICATES	VARCHAR2(4000)	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
FILTER_PREDICATES	VARCHAR2(4000)	Predicates used to filter rows before producing them
PROJECTION	VARCHAR2(4000)	Expressions produced by the operation
TIME	NUMBER	Elapsed time (in seconds) of the operation as estimated by the optimizer's cost-based approach. For statements that use the rule-based approach, this column is null.
QBLOCK_NAME	VARCHAR2(31)	Name of the query block
REMARKS	VARCHAR2(4000)	Remarks
EXECUTIONS	NUMBER	Number of times this cursor has been executed
LAST_STARTS	NUMBER	Number of times this operation has been started, during the last execution
STARTS	NUMBER	Number of times this operation has been started, accumulated over the past executions
LAST_OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, during the last execution
OUTPUT_ROWS	NUMBER	Number of rows produced by the row source, accumulated over the past executions
LAST_CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, during the last execution. Buffers are usually retrieved in consistent mode for queries.
CR_BUFFER_GETS	NUMBER	Number of buffers retrieved in consistent mode, accumulated over the past executions. Buffers are usually retrieved in consistent mode for queries.

Column	Datatype	Description
LAST_CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, during the last execution. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
CU_BUFFER_GETS	NUMBER	Number of buffers retrieved in current mode, accumulated over the past executions. Buffers are retrieved in current mode for statements such as INSERT, UPDATE, and DELETE.
LAST_DISK_READS	NUMBER	Number of physical disk reads performed by the operation, during the last execution
DISK_READS	NUMBER	Number of physical disk reads performed by the operation, accumulated over the past executions
LAST_DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, during the last execution
DISK_WRITES	NUMBER	Number of physical disk writes performed by the operation, accumulated over the past executions
LAST_ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, during the last execution
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) corresponding to this operation, accumulated over the past executions
POLICY	VARCHAR2(10)	Sizing policy for this work area: ■ MANUAL ■ AUTO
ESTIMATED_OPTIMAL_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation completely in memory (optimal execution). This is either derived from optimizer statistics or from previous executions.
ESTIMATED_ONEPASS_SIZE	NUMBER	Estimated size (in KB) required by this work area to execute the operation in a single pass. This is either derived from optimizer statistics or from previous executions.
LAST_MEMORY_USED	NUMBER	Memory size (in KB) used by this work area during the last execution of the cursor
LAST_EXECUTION	VARCHAR2(10)	Indicates whether this work area ran using OPTIMAL, ONE PASS, or under ONE PASS memory requirement (MULTI-PASS), during the last execution of the cursor
LAST_DEGREE	NUMBER	Degree of parallelism used, during the last execution of the cursor
TOTAL_EXECUTIONS	NUMBER	Number of times this work area was active
OPTIMAL_EXECUTIONS	NUMBER	Number of times this work area ran in optimal mode
ONEPASS_EXECUTIONS	NUMBER	Number of times this work area ran in one pass mode
MULTIPASSES_EXECUTIONS	NUMBER	Number of times this work area ran below the one pass memory requirement
ACTIVE_TIME	NUMBER	Average time this work area is active (in hundredths of a second)
MAX_TEMPSEG_SIZE	NUMBER	Maximum temporary segment size (in bytes) created by an instantiation of this work area. This column is null if this work area has never spilled to disk.
LAST_TEMPSEG_SIZE	NUMBER	Temporary segment size (in bytes) created in the last instantiation of this work area. This column is null if the last instantiation of this work area did not spill to disk.

V$SQL_REDIRECTION

V$SQL_REDIRECTION displays SQL statements that are redirected.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the cursor handle
PARENT_HANDLE	RAW(4 | 8)	Address of the parent cursor handle

Column	Datatype	Description
HASH_VALUE	NUMBER	Hash value of the SQL statement
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement
CHILD_NUMBER	NUMBER	Number of the child (instance) for the hash
PARSING_USER_ID	NUMBER	Parsing user ID
PARSING_SCHEMA_ID	NUMBER	Parsing schema ID
COMMAND_TYPE	NUMBER	SELECT, UPDATE, INSERT, MERGE
REASON	VARCHAR2(14)	Reason for redirection ('INVALID OBJECT','ROWID','QUERY REWRITE','READ ONLY')
ERROR_CODE	NUMBER	Error code for local parse
POSITION	NUMBER	Error position, 0 if unknown
SQL_TEXT_PIECE	VARCHAR2(1000)	SQL Text containing position, usually a (qualified) identifier
ERROR_MESSAGE	VARCHAR2(1000)	Error code's corresponding error message resolved in the database language, no arguments resolved

V$SQL_SHARED_CURSOR

V$SQL_SHARED_CURSOR explains why a particular child cursor is not shared with existing child cursors. Each column identifies a specific reason why the cursor cannot be shared.

Column	Datatype	Description
SQL_ID	VARCHAR2(13)	SQL identifier
ADDRESS	RAW(4 | 8)	Address of the parent cursor
CHILD_ADDRESS	RAW(4 | 8)	Address of the child cursor
CHILD_NUMBER	NUMBER	Child number
UNBOUND_CURSOR	VARCHAR2(1)	(Y|N) The existing child cursor was not fully built (in other words, it was not optimized)
SQL_TYPE_MISMATCH	VARCHAR2(1)	(Y|N) The SQL type does not match the existing child cursor
OPTIMIZER_MISMATCH	VARCHAR2(1)	(Y|N) The optimizer environment does not match the existing child cursor
OUTLINE_MISMATCH	VARCHAR2(1)	(Y|N) The outlines do not match the existing child cursor
STATS_ROW_MISMATCH	VARCHAR2(1)	(Y|N) The existing statistics do not match the existing child cursor
LITERAL_MISMATCH	VARCHAR2(1)	(Y|N) Non-data literal values do not match the existing child cursor
FORCE_HARD_PARSE	VARCHAR2(1)	???
SEC_DEPTH_MISMATCH	VARCHAR2(1)	(Y|N) Security level does not match the existing child cursor
EXPLAIN_PLAN_CURSOR	VARCHAR2(1)	(Y|N) The child cursor is an explain plan cursor and should not be shared
BUFFERED_DML_MISMATCH	VARCHAR2(1)	(Y|N) Buffered DML does not match the existing child cursor
PDML_ENV_MISMATCH	VARCHAR2(1)	(Y|N) PDML environment does not match the existing child cursor
INST_DRTLD_MISMATCH	VARCHAR2(1)	(Y|N) Insert direct load does not match the existing child cursor
SLAVE_QC_MISMATCH	VARCHAR2(1)	(Y|N) The existing child cursor is a slave cursor and the new one was issued by the coordinator (or, the existing child cursor was issued by the coordinator and the new one is a slave cursor)
TYPECHECK_MISMATCH	VARCHAR2(1)	(Y|N) The existing child cursor is not fully optimized
AUTH_CHECK_MISMATCH	VARCHAR2(1)	(Y|N) Authorization/translation check failed for the existing child cursor
BIND_MISMATCH	VARCHAR2(1)	(Y|N) The bind metadata does not match the existing child cursor
DESCRIBE_MISMATCH	VARCHAR2(1)	(Y|N) The typecheck heap is not present during the describe for the child cursor

Column	Datatype	Description
LANGUAGE_MISMATCH	VARCHAR2(1)	(Y|N) The language handle does not match the existing child cursor
TRANSLATION_MISMATCH	VARCHAR2(1)	(Y|N) The base objects of the existing child cursor do not match
ROW_LEVEL_SEC_MISMATCH	VARCHAR2(1)	(Y|N) The row level security policies do not match
INSUFF_PRIVS	VARCHAR2(1)	(Y|N) Insufficient privileges on objects referenced by the existing child cursor
INSUFF_PRIVS_REM	VARCHAR2(1)	(Y|N) Insufficient privileges on remote objects referenced by the existing child cursor
REMOTE_TRANS_MISMATCH	VARCHAR2(1)	(Y|N) The remote base objects of the existing child cursor do not match
LOGMINER_SESSION_MISMATCH	VARCHAR2(1)	(Y|N) LogMiner Session parameters mismatch
INCOMP_LTRL_MISMATCH	VARCHAR2(1)	(Y|N) Cursor might have some binds (literals) which may be unsafe/non-data. Value mismatch.
OVERLAP_TIME_MISMATCH	VARCHAR2(1)	Mismatch caused by setting session parameter ERROR_ON_OVERLAP_TIME
EDITIONS_MISMATCH	VARCHAR2(1)	???
SQL_REDIRECT_MISMATCH	VARCHAR2(1)	SQL redirection mismatch
MV_QUERY_GEN_MISMATCH	VARCHAR2(1)	Internal, used to force a hard-parse when analyzing materialized view queries
USER_BIND_PEEK_MISMATCH	VARCHAR2(1)	Cursor is not shared because value of one or more user binds is different and this has a potential to change the execution plan
TYPCHK_DEP_MISMATCH	VARCHAR2(1)	Cursor has typecheck dependencies
NO_TRIGGER_MISMATCH	VARCHAR2(1)	Cursor and child have no trigger mismatch
FLASHBACK_CURSOR	VARCHAR2(1)	Cursor non-shareability due to flashback
ANYDATA_TRANSFORMATION	VARCHAR2(1)	Is criteria for opaque type transformation and does not match
INCOMPLETE_CURSOR	VARCHAR2(1)	Cursor is incomplete: typecheck heap came from call memory
TOP_LEVEL_RPI_CURSOR	VARCHAR2(1)	Is top level RPI cursor
DIFFERENT_LONG_LENGTH	VARCHAR2(1)	Value of LONG does not match
LOGICAL_STANDBY_APPLY	VARCHAR2(1)	Logical standby apply context does not match
DIFF_CALL_DURN	VARCHAR2(1)	If Slave SQL cursor/single call
BIND_UACS_DIFF	VARCHAR2(1)	One cursor has bind UACs and one does not
PLSQL_CMP_SWITCHS_DIFF	VARCHAR2(1)	PL/SQL anonymous block compiled with different PL/SQL compiler switches
CURSOR_PARTS_MISMATCH	VARCHAR2(1)	Cursor was compiled with subexecution (cursor parts were executed)
STB_OBJECT_MISMATCH	VARCHAR2(1)	STB has come into existence since cursor was compiled
EDITION_FOR_TRIGGERS_MISMATCH	VARCHAR2(1)	???
PQ_SLAVE_MISMATCH	VARCHAR2(1)	Top-level slave decides not to share cursor
TOP_LEVEL_DDL_MISMATCH	VARCHAR2(1)	Is top-level DDL cursor
MULTI_PX_MISMATCH	VARCHAR2(1)	Cursor has multiple parallelizers and is slave-compiled
BIND_PEEKED_PQ_MISMATCH	VARCHAR2(1)	Cursor based around bind peeked values
MV_REWRITE_MISMATCH	VARCHAR2(1)	Cursor needs recompilation because an SCN was used during compile time due to being rewritten by materialized view
ROLL_INVALID_MISMATCH	VARCHAR2(1)	Marked for rolling invalidation and invalidation window exceeded

Column	Datatype	Description
OPTIMIZER_MODE_ MISMATCH	VARCHAR2(1)	Parameter OPTIMIZER_MODE mismatch (for example, all_rows versus first_rows_1)
PX_MISMATCH	VARCHAR2(1)	Mismatch in one parameter affecting the parallelization of a SQL statement. For example, one cursor was compiled with parallel DML enabled while the other was not.
MV_STALEOBJ_MISMATCH	VARCHAR2(1)	Cursor cannot be shared because there is a mismatch in the list of materialized views which were stale at the time the cursor was built
FLASHBACK_TABLE_ MISMATCH	VARCHAR2(1)	Cursor cannot be shared because there is a mismatch with triggers being enabled and/or referential integrity constraints being deferred
LITREP_COMP_MISMATCH	VARCHAR2(1)	Mismatch in use of literal replacement

V$SQL_SHARED_MEMORY

V$SQL_SHARED_MEMORY displays information about the cursor shared memory snapshot. Each SQL statement stored in the shared pool has one or more child objects associated with it. Each child object has a number of parts, one of which is the context heap, which holds, among other things, the query plan.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	SQL text of the shared cursor child object for which this row is displaying information
SQL_FULLTEXT	CLOB	Full text for the SQL statement exposed as a CLOB column. The full text of a SQL statement can be retrieved using this column instead of joining with the V$SQLTEXT dynamic performance view.
HASH_VALUE	NUMBER	Hash value of the above SQL text in the shared pool
SQL_ID	VARCHAR2(13)	SQL identifier of the above SQL text in the shared pool
HEAP_DESC	RAW(4 | 8)	Address of the descriptor for the context heap of the child cursor described in this row
STRUCTURE	VARCHAR2(16)	If the memory chunk described in this row was allocated using a comment of the form "X : Y", then this is the "X" part of the comment
FUNCTION	VARCHAR2(16)	Similar to the STRUCTURE column, this is the "Y" field of the comment
CHUNK_COM	VARCHAR2(16)	Whole comment field that was supplied when this memory chunk was allocated
CHUNK_PTR	RAW(4 | 8)	Starting address of the allocated memory chunk
CHUNK_SIZE	NUMBER	Amount of memory allocated for this chunk
ALLOC_CLASS	VARCHAR2(8)	Class of memory that this chunk of memory belongs to. It will usually be either FREEABLE or PERMANENT.
CHUNK_TYPE	NUMBER	An index into a table of callback functions that tell the server how to re-create this chunk of memory should it need to be removed from the shared pool based on an LRU algorithm
SUBHEAP_DESC	RAW(4 | 8)	If the parent heap of this context heap is itself a subheap, then this is the address of the descriptor of the parent heap

V$SQL_WORKAREA

V$SQL_WORKAREA displays information about work areas used by SQL cursors. Each SQL statement stored in the shared pool has one or more child cursors that are listed in the V$SQL view. V$SQL_WORKAREA lists all work areas needed by these child cursors; V$SQL_WORKAREA can be joined with V$SQLAREA on (ADDRESS, HASH_VALUE) and with V$SQL on (ADDRESS, HASH_VALUE, CHILD_NUMBER).

You can use this view to find out answers to the following questions:

- What are the top 10 work areas that require the most cache area?
- For work areas allocated in AUTO mode, what percentage of work areas are running using maximum memory?

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Address of the parent cursor handle
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache. Two columns PARENT_HANDLE and HASH_VALUE can be used to join with V$SQLAREA to locate the parent cursor.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent statement in the library cache
CHILD_NUMBER	NUMBER	Number of the child cursor that uses this work area. The columns PARENT_HANDLE, HASH_VALUE, and CHILD_NUMBER can be used to join with V$SQL to locate the child cursor using this area.
WORKAREA_ADDRESS	RAW(4 | 8)	Address of the work area handle. This is the primary key for the view.
OPERATION_TYPE	VARCHAR2(20)	Type of operation using the work area (SORT, HASH JOIN, GROUP BY, BUFFERING, BITMAP MERGE, or BITMAP CREATE)
OPERATION_ID	NUMBER	A unique number used to identify the operation in the execution plan. This identifier can be joined to V$SQL_PLAN to locate the operation that uses this work area.
POLICY	VARCHAR2(10)	Sizing policy for this work area (MANUAL or AUTO)
ESTIMATED_OPTIMAL_SIZE	NUMBER	Estimated size (in bytes) required by this work area to execute the operation completely in memory (optimal execution). Derived from either optimizer statistics or previous executions.
ESTIMATED_ONEPASS_SIZE	NUMBER	Estimated size (in bytes) required by this work area to execute the operation in a single pass. Derived from either optimizer statistics or previous executions.
LAST_MEMORY_USED	NUMBER	Memory (in bytes) used by this work area during the last execution of the cursor
LAST_EXECUTION	VARCHAR2(10)	Indicates whether this work area runs using OPTIMAL, ONE PASS, or ONE PASS memory requirement (or MULTI-PASS), during the last execution of the cursor
LAST_DEGREE	NUMBER	Degree of parallelism used during the last execution of this operation
TOTAL_EXECUTIONS	NUMBER	Number of times this work area was active
OPTIMAL_EXECUTIONS	NUMBER	Number of times this work area ran in optimal mode
ONEPASS_EXECUTIONS	NUMBER	Number of times this work area ran in one-pass mode
MULTIPASSES_EXECUTIONS	NUMBER	Number of times this work area ran below the one-pass memory requirement
ACTIVE_TIME	NUMBER	Average time this work area is active (in hundredths of a second)
MAX_TEMPSEG_SIZE	NUMBER	Maximum temporary segment size (in bytes) created by an instantiation of this work area. This column is NULL if this work area has never spilled to disk.
LAST_TEMPSEG_SIZE	NUMBER	Temporary segment size (in bytes) created in the last instantiation of this work area. This column is NULL if the last instantiation of this work area did not spill to disk.

V$SQL_WORKAREA_ACTIVE

V$SQL_WORKAREA_ACTIVE contains an instantaneous view of the work areas currently allocated by the system. You can join this view against V$SQL_WORKAREA on WORKAREA_ADDRESS to access the definition of that work area. If a work area spills to disk, then this view contains information for the temporary segment created on behalf of this work area.

The last three columns are included to enable joining V$SQL_WORKAREA_ACTIVE with V$TEMPSEG_USAGE to retrieve more information on this temporary segment.

You can use this view to answer the following:

- What are the top 10 largest work areas currently allocated in the system?
- What percentage of memory is over-allocated (EXPECTED_SIZE < ACTUAL_MEM_USED) and under-allocated (EXPECTED_SIZE > ACTUAL_MEM_USED)?
- What are the active work areas using more memory than what is expected by the memory manager?
- What are the active work areas that have spilled to disk?

Column	Datatype	Description
SQL_HASH_VALUE	NUMBER	Hash value of the SQL statement that is currently being executed
SQL_ID	VARCHAR2(13)	SQL identifier of the SQL statement that is currently being executed
SQL_EXEC_START	DATE	Time when the execution of the SQL currently executed by this session started
SQL_EXEC_ID	NUMBER	SQL execution identifier (see V$SQL_MONITOR)
WORKAREA_ADDRESS	RAW(4 | 8)	Address of the work area handle. This is the primary key for the view.
OPERATION_TYPE	VARCHAR2(20)	Type of operation using the work area (SORT, HASH JOIN, GROUP BY, BUFFERING, BITMAP MERGE, or BITMAP CREATE)
OPERATION_ID	NUMBER	A unique number used to identify the operation in the execution plan. This identifier can be joined to V$SQL_PLAN to locate the operation that uses this work area.
POLICY	VARCHAR2(6)	Sizing policy for this work area (MANUAL or AUTO)
SID	NUMBER	Session identifier
QCINST_ID	NUMBER	Query coordinator instance identifier. Along with QCSID, enables you to uniquely identify the query coordinator.
QCSID	NUMBER	Query coordinator session identifier. This is the same as the SID if the work area is allocated by a serial cursor.
ACTIVE_TIME	NUMBER	Average time this work area is active (in hundredths of a second)
WORK_AREA_SIZE	NUMBER	Maximum size (in bytes) of the work area as it is currently used by the operation
EXPECTED_SIZE	NUMBER	Expected size (in bytes) for this work area. EXPECTED_SIZE is set on behalf of the operation by the memory manager. Memory can be over-allocated when WORK_AREA_SIZE has a higher value than EXPECTED_SIZE. This can occur when the operation using this work area takes a long time to resize it.
ACTUAL_MEM_USED	NUMBER	Amount of PGA memory (in bytes) currently allocated on behalf of this work area. This value should range between 0 and WORK_AREA_SIZE.
MAX_MEM_USED	NUMBER	Maximum memory amount (in bytes) used by this work area
NUMBER_PASSES	NUMBER	Number of passes corresponding to this work area (0 if running in OPTIMAL mode)
TEMPSEG_SIZE	NUMBER	Size (in bytes) of the temporary segment used on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
TABLESPACE	VARCHAR2(30)	Tablespace name for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
SEGRFNO#	NUMBER	Relative file number within the tablespace for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.
SEGBLK#	NUMBER	Block number for the temporary segment created on behalf of this work area. This column is NULL if this work area has not (yet) spilled to disk.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on how to monitor SQL work areas

V$SQL_WORKAREA_HISTOGRAM

V$SQL_WORKAREA_HISTOGRAM displays the cumulative work area execution statistics (cumulated since instance startup) for different work area groups. The work areas are split into 33 groups based on their optimal memory requirements with the requirements increasing in powers of two. That is, work areas whose optimal requirement varies from 0 KB to 1 KB, 1 KB to 2 KB, 2 KB to 4 KB, ... and 2 TB to 4 TB.

For each work area group, the V$SQL_WORKAREA_HISTOGRAM view shows how many work areas in that group were able to run in optimal mode, how many were able to run in one-pass mode, and finally how many ran in multi-pass mode. The DBA can take a snapshot at the beginning and the end of a desired time interval to derive the same statistics for that interval.

Column	Datatype	Description
LOW_OPTIMAL_SIZE	NUMBER	Lower bound for the optimal memory requirement of work areas included in this row (bytes)
HIGH_OPTIMAL_SIZE	NUMBER	Upper bound for the optimal memory requirement of work areas included in this row (bytes)
OPTIMAL_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in optimal mode since instance startup
ONEPASS_EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in one-pass mode since instance startup
MULTIPASSES_ EXECUTIONS	NUMBER	Number of work areas with an optimal memory requirement comprised between LOW_OPTIMAL_SIZE and HIGH_OPTIMAL_SIZE which have been executed in multi-pass mode since instance startup
TOTAL_EXECUTIONS	NUMBER	Sum of OPTIMAL_EXECUTIONS, ONEPASS_EXECUTIONS, and MULTIPASSES_EXECUTIONS

> **See Also:** *Oracle Database Performance Tuning Guide* for detailed information on how to monitor automatic PGA memory performance using this view

V$SQLAREA

V$SQLAREA displays statistics on shared SQL areas and contains one row per SQL string. It provides statistics on SQL statements that are in memory, parsed, and ready for execution.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor
SQL_FULLTEXT	CLOB	All characters of the SQL text for the current cursor
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
SHARABLE_MEM	NUMBER	Amount of shared memory used by a cursor. If multiple child cursors exist, then the sum of all shared memory used by all child cursors.
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of an open cursor. If multiple child cursors exist, then the fixed sum of memory used for the lifetime of all the child cursors.

Column	Datatype	Description
RUNTIME_MEM	NUMBER	Fixed amount of memory required during execution of a cursor. If multiple child cursors exist, then the fixed sum of all memory required during execution of all the child cursors.
SORTS	NUMBER	Sum of the number of sorts that were done for all the child cursors
VERSION_COUNT	NUMBER	Number of child cursors that are present in the cache under this parent
LOADED_VERSIONS	NUMBER	Number of child cursors that are present in the cache and have their context heap loaded
OPEN_VERSIONS	NUMBER	Number of child cursors that are currently open under this current parent
USERS_OPENING	NUMBER	Number of users that have any of the child cursors open
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Total number of executions, totalled over all the child cursors
PX_SERVERS_EXECUTIONS	NUMBER	Total number of executions performed by parallel execution servers (0 when the statement has never been executed in parallel)
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
USERS_EXECUTING	NUMBER	Total number of users executing the statement over all child cursors
LOADS	NUMBER	Number of times the object was loaded or reloaded
FIRST_LOAD_TIME	VARCHAR2(19)	Timestamp of the parent creation time
INVALIDATIONS	NUMBER	Total number of invalidations over all the child cursors
PARSE_CALLS	NUMBER	Sum of all parse calls to all the child cursors under this parent
DISK_READS	NUMBER	Sum of the number of disk reads over all child cursors
DIRECT_WRITES	NUMBER	Sum of the number of direct writes over all child cursors
BUFFER_GETS	NUMBER	Sum of buffer gets over all child cursors
APPLICATION_WAIT_TIME	NUMBER	Application wait time (in microseconds)
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time (in microseconds)
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time (in microseconds)
USER_IO_WAIT_TIME	NUMBER	User I/O Wait Time (in microseconds)
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time (in microseconds)
JAVA_EXEC_TIME	NUMBER	Java execution time (in microseconds)
ROWS_PROCESSED	NUMBER	Total number of rows processed on behalf of this SQL statement
COMMAND_TYPE	NUMBER	Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement was executed
OPTIMIZER_COST	NUMBER	Cost of this query given by the optimizer
OPTIMIZER_ENV	RAW(2000)	Optimizer environment
OPTIMIZER_ENV_HASH_VALUE	NUMBER	Hash value for the optimizer environment
PARSING_USER_ID	NUMBER	User ID of the user that has parsed the very first cursor under this parent
PARSING_SCHEMA_ID	NUMBER	Schema ID that was used to parse this child cursor
PARSING_SCHEMA_NAME	VARCHAR2(30)	Schema name that was used to parse this child cursor
KEPT_VERSIONS	NUMBER	Number of child cursors that have been marked to be kept using the DBMS_SHARED_POOL package
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache

Column	Datatype	Description
OLD_HASH_VALUE	NUMBER	Old SQL hash value
PLAN_HASH_VALUE	NUMBER	Numeric representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	Hash value of the module that is named in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	Hash value of the action that is named in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction failed to serialize, producing ORA-08177 errors, totalled over all the child cursors
OUTLINE_CATEGORY	VARCHAR2(64)	If an outline was applied during construction of the cursor, then this column displays the category of that outline. Otherwise the column is left blank.
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing, executing, and fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing, executing, and fetching
OUTLINE_SID	VARCHAR2(40)	Outline session identifier
LAST_ACTIVE_CHILD_ADDRESS	RAW(4 | 8)	Address (identifier) of the child cursor that was the last to be active in the group (that is, the child cursor on behalf of which statistics in V$SQL were updated)
REMOTE	VARCHAR2(1)	Indicates whether the cursor is remote mapped (Y) or not (N)
OBJECT_STATUS	VARCHAR2(19)	Status of the cursor: ▪ VALID - Valid, authorized without errors ▪ VALID_AUTH_ERROR - Valid, authorized with authorization errors ▪ VALID_COMPILE_ERROR - Valid, authorized with compilation errors ▪ VALID_UNAUTH - Valid, unauthorized ▪ INVALID_UNAUTH - Invalid, unauthorized ▪ INVALID - Invalid, unauthorized but keep the timestamp
LITERAL_HASH_VALUE	NUMBER	Hash value of the literals which are replaced with system-generated bind variables and are to be matched, when CURSOR_SHARING is used. This is not the hash value for the SQL statement. If CURSOR_SHARING is not used, then the value is 0.
LAST_LOAD_TIME	DATE	Time at which the query plan was loaded into the library cache
IS_OBSOLETE	VARCHAR2(1)	Indicates whether the cursor has become obsolete (Y) or not (N). This can happen if the number of child cursors is too large.
IS_BIND_SENSITIVE	VARCHAR2(1)	Indicates whether the cursor is bind sensitive (Y) or not (N). A query is considered bind-sensitive if the optimizer peeked at one of its bind variable values when computing predicate selectivities and where a change in a bind variable value may cause the optimizer to generate a different plan.
IS_BIND_AWARE	VARCHAR2(1)	Indicates whether the cursor is bind aware (Y) or not (N). A query is considered bind-aware if it has been marked to use extended cursor sharing. The query would already have been marked as bind-sensitive.
IS_SHAREABLE	VARCHAR2(1)	Indicates whether the cursor can be shared (Y) or not (N)
CHILD_LATCH	NUMBER	Child latch number that is protecting the cursor
SQL_PROFILE	VARCHAR2(64)	SQL profile used for this statement, if any
SQL_PATCH	VARCHAR2(30)	SQL patch used for this statement, if any
SQL_PLAN_BASELINE	VARCHAR2(30)	SQL plan baseline used for this statement, if any

Column	Datatype	Description
PROGRAM_ID	NUMBER	Program identifier
PROGRAM_LINE#	NUMBER	Program line number
EXACT_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to EXACT
FORCE_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to FORCE
LAST_ACTIVE_TIME	DATE	Time at which the query plan was last active
BIND_DATA	RAW(2000)	Bind data
TYPECHECK_MEM	NUMBER	???

V$SQLAREA_PLAN_HASH

V$SQLAREA_PLAN_HASH displays statistics on shared SQL area (V$SQL) by grouping on the SQL_ID and PLAN_HASH_VALUE columns. It can potentially create several rows for one parent cursor, one for each distinct value of the column PLAN_HASH_VALUE.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor
SQL_FULLTEXT	CLOB	All characters of the SQL text for the current cursor
ADDRESS	RAW(4 | 8)	Address of the handle to the parent for this cursor
HASH_VALUE	NUMBER	Hash value of the parent statement in the library cache
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
PLAN_HASH_VALUE	NUMBER	Numeric representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
VERSION_COUNT	NUMBER	Number of child cursors that are present in the cache under this parent
LAST_ACTIVE_CHILD_ADDRESS	RAW(4 | 8)	Address (identifier) of the child cursor that was the last to be active in the group (that is, the child cursor on behalf of which statistics in V$SQL were updated)
SHARABLE_MEM	NUMBER	Amount of shared memory used by a cursor. If multiple child cursors exist, then it is the sum of all shared memory used by all child cursors.
PERSISTENT_MEM	NUMBER	Fixed amount of memory used for the lifetime of an open cursor. If multiple child cursors exist, then it is the fixed sum of memory used for the lifetime of all the child cursors.
RUNTIME_MEM	NUMBER	Fixed amount of memory required during execution of a cursor. If multiple child cursors exist, then the fixed sum of all memory required during execution of all the child cursors.
SORTS	NUMBER	Sum of the number of sorts that were done for all the child cursors
LOADED_VERSIONS	NUMBER	Number of child cursors that are present in the cache and that have their context heap loaded
OPEN_VERSIONS	NUMBER	Number of child cursors that are currently open under this parent
USERS_OPENING	NUMBER	Number of users that have any of the child cursors open
USERS_EXECUTING	NUMBER	Total number of users executing the statement over all child cursors
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Total number of executions, totalled over all the child cursors
PX_SERVERS_EXECUTIONS	NUMBER	Total number of executions performed by parallel execution servers (0 when the statement has never been executed in parallel)

Column	Datatype	Description
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
LOADS	NUMBER	Number of times the object was loaded or reloaded
FIRST_LOAD_TIME	DATE	Timestamp of the parent creation time
LAST_LOAD_TIME	DATE	Time at which the query plan was loaded into the library cache
LAST_ACTIVE_TIME	DATE	TIme at which the query plan was last active
INVALIDATIONS	NUMBER	Total number of invalidations over all the child cursors
PARSE_CALLS	NUMBER	Sum of all parse calls to all the child cursors under this parent
DISK_READS	NUMBER	Sum of the number of disk reads over all child cursors
DIRECT_WRITES	NUMBER	Sum of the number of direct writes over all child cursors
BUFFER_GETS	NUMBER	Sum of buffer gets over all child cursors
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing, executing, and fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing, executing, and fetching
APPLICATION_WAIT_TIME	NUMBER	Application wait time (in microseconds)
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time (in microseconds)
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time (in microseconds)
USER_IO_WAIT_TIME	NUMBER	User I/O wait time (in microseconds)
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time (in microseconds)
JAVA_EXEC_TIME	NUMBER	Java execution time (in microseconds)
ROWS_PROCESSED	NUMBER	Total number of rows processed on behalf of this SQL statement
COMMAND_TYPE	NUMBER	Oracle command type definition
OPTIMIZER_MODE	VARCHAR2(10)	Mode under which the SQL statement was executed
OPTIMIZER_COST	NUMBER	Cost of this query given by the optimizer
OPTIMIZER_ENV	RAW(2000)	Optimizer environment
OPTIMIZER_ENV_HASH_VALUE	NUMBER	Hash value for the optimizer environment
PARSING_USER_ID	NUMBER	User ID of the user that has parsed the very first cursor under this parent
PARSING_SCHEMA_ID	NUMBER	Schema ID that was used to parse this child cursor
PARSING_SCHEMA_NAME	VARCHAR2(30)	Schema name that was used to parse this child cursor
KEPT_VERSIONS	NUMBER	Number of child cursors that have been marked to be kept using the DBMS_SHARED_POOL package
MODULE	VARCHAR2(64)	Contains the name of the module that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_MODULE
MODULE_HASH	NUMBER	Hash value of the module that is named in the MODULE column
ACTION	VARCHAR2(64)	Contains the name of the action that was executing at the time that the SQL statement was first parsed as set by calling DBMS_APPLICATION_INFO.SET_ACTION
ACTION_HASH	NUMBER	Hash value of the action that is named in the ACTION column
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction failed to serialize, producing ORA-08177 errors, totalled over all the child cursors

Column	Datatype	Description
OUTLINE_CATEGORY	VARCHAR2(64)	If an outline was applied during construction of the cursor, then this column displays the category of that outline. Otherwise, the column is left blank.
OUTLINE_SID	VARCHAR2(40)	Outline session identifier
REMOTE	VARCHAR2(1)	Indicates whether the cursor is remote mapped (Y) or not (N)
OBJECT_STATUS	VARCHAR2(19)	Status of the cursor: ■ VALID - Valid, authorized without errors ■ VALID_AUTH_ERROR - Valid, authorized with authorization errors ■ VALID_COMPILE_ERROR - Valid, authorized with compilation errors ■ VALID_UNAUTH - Valid, unauthorized ■ INVALID_UNAUTH - Invalid, unauthorized ■ INVALID - Invalid, unauthorized but keep the timestamp
LITERAL_HASH_VALUE	NUMBER	Hash value of the literals which are replaced with system-generated bind variables and are to be matched, when CURSOR_SHARING is used. This is not the hash value for the SQL statement. If CURSOR_SHARING is not used, then the value is 0.
SQL_PROFILE	VARCHAR2(64)	SQL profile used for this statement, if any
PROGRAM_ID	NUMBER	Program identifier
PROGRAM_LINE#	NUMBER	Program line number
EXACT_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to EXACT
FORCE_MATCHING_SIGNATURE	NUMBER	Signature used when the CURSOR_SHARING parameter is set to FORCE
BIND_DATA	RAW(2000)	Bind data
TYPECHECK_MEM	NUMBER	???

V$SQLFN_ARG_METADATA

V$SQLFN_ARG_METADATA contains metadata about function arguments. There is one row for each argument of every function found in V$SQLFN_METADATA. There are no rows for functions that do not have any arguments.

Column	Datatype	Description
FUNC_ID	NUMBER	Internal function identification number. This column can be used to join with the V$SQLFN_METADATA view.
ARGNUM	NUMBER	Argument number
DATATYPE	VARCHAR2(8)	Datatype of the argument. The value is NULL if this argument is not used. Otherwise, it can take values of any Oracle datatype, family datatype, or EXPR datatype.
DESCR	VARCHAR2(30)	This column is reserved for future use.

V$SQLFN_METADATA

V$SQLFN_METADATA contains metadata about operators and built-in functions. Note that this view does not contain information about arguments because the number of arguments will be different for various functions. Information about arguments is contained in V$SQLFN_ARG_METADATA, which can be joined with V$SQLFN_METADATA to get information about any function and its arguments.

Column	Datatype	Description
FUNC_ID	NUMBER	Internal function identification number
NAME	VARCHAR2(30)	Name of the built-in function
MINARGS	NUMBER	Minimum number of arguments for the function
MAXARGS	NUMBER	Maximum number of arguments for the function
DATATYPE	VARCHAR2(8)	Return datatype, which can take any Oracle datatype values, datatype family values, or ARG[n] datatypes
VERSION	VARCHAR2(12)	Minimum version of the database that has this function
ANALYTIC	VARCHAR2(3)	Indicates whether the function is an analytic function(YES) or not (NO)
AGGREGATE	VARCHAR2(3)	Indicates whether the function is an aggregate function(YES) or not (NO)
DISP_TYPE	VARCHAR2(13)	Function display type: ■ NORMAL ■ ARITHMETIC ■ PARENTHESIS ■ REL-OP ■ CASELIKE ■ NOPARENTHESIS
USAGE	VARCHAR2(30)	A text explanation of how to use this function. The text is based on the syntax diagram for the function in the *Oracle Database SQL Language Reference*.
DESCR	VARCHAR2(30)	Description of the function

V$SQLSTATS

V$SQLSTATS displays basic performance statistics for SQL cursors, with each row representing the data for a unique combination of SQL text and optimizer plan (that is, unique combination of SQL_ID, and PLAN_HASH_VALUE). The column definitions for columns in V$SQLSTATS are identical to those in the V$SQL and V$SQLAREA views. However, the V$SQLSTATS view differs from V$SQL and V$SQLAREA in that it is faster, more scalable, and has a greater data retention (the statistics may still appear in this view, even after the cursor has been aged out of the shared pool). Note that V$SQLSTATS contains a subset of columns that appear in V$SQL and V$SQLAREA.

Column	Datatype	Description
SQL_TEXT	VARCHAR2(1000)	First thousand characters of the SQL text for the current cursor
SQL_FULLTEXT	CLOB	Full text for the SQL statement exposed as a CLOB column. THe full text of a SQL statement can be retrieved using this column instead of joining with the V$SQLTEXT view.
SQL_ID	VARCHAR2(13)	SQL identifier of the parent cursor in the library cache
LAST_ACTIVE_TIME	DATE	Last time that statistics for a cursor with this text and plan were updated
LAST_ACTIVE_CHILD_ADDRESS	RAW(4 | 8)	Address of the cursor that last updated statistics
PLAN_HASH_VALUE	NUMBER	Numeric representation of the SQL plan for this cursor. Comparing one PLAN_HASH_VALUE to another easily identifies whether or not two plans are the same (rather than comparing the two plans line by line).
PARSE_CALLS	NUMBER	Number of parse calls for all cursors with this SQL text and plan
DISK_READS	NUMBER	Number of disk reads for all cursors with this SQL text and plan
DIRECT_WRITES	NUMBER	Number of direct writes for all cursors with this SQL text and plan
BUFFER_GETS	NUMBER	Number of buffer gets for all cursors with this SQL text and plan
ROWS_PROCESSED	NUMBER	Total number of rows the parsed SQL statement returns

Column	Datatype	Description
SERIALIZABLE_ABORTS	NUMBER	Number of times the transaction failed to serialize, producing ORA-08177 errors, per cursor
FETCHES	NUMBER	Number of fetches associated with the SQL statement
EXECUTIONS	NUMBER	Number of executions that took place on this object since it was brought into the library cache
END_OF_FETCH_COUNT	NUMBER	Number of times this cursor was fully executed since the cursor was brought into the library cache. The value of this statistic is not incremented when the cursor is partially executed, either because it failed during the execution or because only the first few rows produced by this cursor are fetched before the cursor is closed or re-executed. By definition, the value of the END_OF_FETCH_COUNT column should be less or equal to the value of the EXECUTIONS column.
LOADS	NUMBER	Number of times the object was either loaded or reloaded
VERSION_COUNT	NUMBER	number of cursors present in the cache with this SQL text and plan
INVALIDATIONS	NUMBER	Number of times this child cursor has been invalidated
PX_SERVERS_EXECUTIONS	NUMBER	Total number of executions performed by parallel execution servers (0 when the statement has never been executed in parallel)
CPU_TIME	NUMBER	CPU time (in microseconds) used by this cursor for parsing, executing, and fetching
ELAPSED_TIME	NUMBER	Elapsed time (in microseconds) used by this cursor for parsing, executing, and fetching
AVG_HARD_PARSE_TIME	NUMBER	Average hard parse time (in microseconds) used by this cursor for parsing, executing, and fetching
APPLICATION_WAIT_TIME	NUMBER	Application wait time (in microseconds)
CONCURRENCY_WAIT_TIME	NUMBER	Concurrency wait time (in microseconds)
CLUSTER_WAIT_TIME	NUMBER	Cluster wait time (in microseconds)
USER_IO_WAIT_TIME	NUMBER	User I/O wait time (in microseconds)
PLSQL_EXEC_TIME	NUMBER	PL/SQL execution time (in microseconds)
JAVA_EXEC_TIME	NUMBER	Java execution time (in microseconds)
SORTS	NUMBER	Number of sorts that were done for the child cursor
SHARABLE_MEM	NUMBER	Total shared memory (in bytes) currently occupied by all cursors with this SQL text and plan
TOTAL_SHARABLE_MEM	NUMBER	Total shared memory (in bytes) occupied by all cursors with this SQL text and plan if they were to be fully loaded in the shared pool (that is, cursor size)
TYPECHECK_MEM	NUMBER	???

V$SQLTEXT

V$SQLTEXT displays the text of SQL statements belonging to shared SQL cursors in the SGA.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Used with HASH_VALUE to uniquely identify a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify a cached cursor
SQL_ID	VARCHAR2(13)	SQL identifier of a cached cursor
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2(64)	A column containing one piece of the SQL text

V$SQLTEXT_WITH_NEWLINES

V$SQLTEXT_WITH_NEWLINES is identical to the V$SQLTEXT view except that, to improve legibility, V$SQLTEXT_WITH_NEWLINES does not replace newlines and tabs in the SQL statement with spaces.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Used with HASH_VALUE to uniquely identify a cached cursor
HASH_VALUE	NUMBER	Used with ADDRESS to uniquely identify a cached cursor
SQL_ID	VARCHAR2(13)	SQL identifier of a cached cursor
COMMAND_TYPE	NUMBER	Code for the type of SQL statement (SELECT, INSERT, and so on)
PIECE	NUMBER	Number used to order the pieces of SQL text
SQL_TEXT	VARCHAR2(64)	A column containing one piece of the SQL text

See Also: "V$SQLTEXT" on page 8-53

V$SSCR_SESSIONS

V$SSCR_SESSIONS ???.

Column	Datatype	Description
SID	NUMBER	???
SERIAL#	NUMBER	???
STATE	VARCHAR2(9)	???
CRMODE	VARCHAR2(7)	???
SCOPE	VARCHAR2(7)	???
NC_COMPONENT	VARCHAR2(40)	???
NC_REASON	NUMBER	???
OPTIONS	NUMBER	???
TIMEOUT	NUMBER	???

V$STANDBY_LOG

V$STANDBY_LOG displays information about standby redo logs. Standby redo logs are similar to online redo logs, but standby redo logs are only used on a standby database that is receiving redo data from the primary database.

Column	Datatype	Description
GROUP#	NUMBER	Log group number
DBID	VARCHAR2(40)	Database ID of the primary database to which the standby redo logfile is assigned. If the standby redo logfile is unassigned, the value UNASSIGNED will be displayed.
THREAD#	NUMBER	Log thread number
SEQUENCE#	NUMBER	Log sequence number
BYTES	NUMBER	Size of the log (in bytes)
USED	NUMBER	Number of bytes used in the log
ARCHIVED	VARCHAR2(3)	Archive status (YES) or (NO). See the STATUS column for further details.

Column	Datatype	Description
STATUS	VARCHAR2(10)	Log status: - UNASSIGNED - If ARCHIVED is NO, then the standby redo log has been archived and is again available. If ARCHIVED is YES, then the standby redo log has never been used and is available. - ACTIVE - If ARCHIVED is NO, then the standby redo log is complete and waiting to be archived. If ARCHIVED is YES, then the standby redo log is currently being written to and is therefore not ready to be archived. For a given thread, there should be only one such log.
FIRST_CHANGE#	NUMBER	Lowest SCN in the log
FIRST_TIME	DATE	Time of the first SCN in the log
LAST_CHANGE#	NUMBER	Last change number made to this datafile. Set to NULL if the datafile is being changed.
LAST_TIME	DATE	Timestamp of the last change

V$STATISTICS_LEVEL

V$STATISTICS_LEVEL displays the status of the statistics/advisories controlled by STATISTICS_LEVEL.

Column	Datatype	Description
STATISTICS_NAME	VARCHAR2(64)	Name of the statistic/advisory
DESCRIPTION	VARCHAR2(4000)	Description of the statistic/advisory
SESSION_STATUS	VARCHAR2(8)	Status of the statistic/advisory for the session: - ENABLED - DISABLED
SYSTEM_STATUS	VARCHAR2(8)	System-wide status of the statistic/advisory: - ENABLED - DISABLED
ACTIVATION_LEVEL	VARCHAR2(7)	Indicates the level of STATISTICS_LEVEL that enables the statistic/advisory: - BASIC - TYPICAL - ALL
STATISTICS_VIEW_NAME	VARCHAR2(64)	If there is a single view externalizing the statistic/advisory, then this column contains the name of that view. If there is no such view, then this column is null. If there are multiple views involved, then the DESCRIPTION column mentions the view names.
SESSION_SETTABLE	VARCHAR2(3)	Indicates whether the statistic/advisory can be set at the session level (YES) or not (NO)

See Also: "STATISTICS_LEVEL" on page 1-157

V$STATNAME

V$STATNAME displays decoded statistic names for the statistics shown in the V$SESSTAT and V$SYSSTAT tables.

On some platforms, the NAME and CLASS columns contain additional operating system-specific statistics.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number **Note:** Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2(64)	Statistic name
CLASS	NUMBER	A number representing one or more statistics classes. The following class numbers are additive: ■ 1 - User ■ 2 - Redo ■ 4 - Enqueue ■ 8 - Cache ■ 16 - OS ■ 32 - Real Application Clusters ■ 64 - SQL ■ 128 - Debug
STAT_ID	NUMBER	Identifier of the statistic

See Also:

- "V$SESSTAT" on page 8-20 and "V$SYSSTAT" on page 8-67
- Appendix E, "Statistics Descriptions" for a description of each statistic
- Your operating system-specific Oracle documentation

V$STREAMS_APPLY_COORDINATOR

V$STREAMS_APPLY_COORDINATOR displays information about each apply process coordinator. The coordinator for an apply process gets transactions from the apply process reader and passes them to apply servers.

Column	Datatype	Description
SID	NUMBER	Session ID of the coordinator's session
SERIAL#	NUMBER	Serial number of the coordinator's session
STATE	VARCHAR2(21)	State of the coordinator: ■ INITIALIZING ■ APPLYING ■ SHUTTING DOWN CLEANLY ■ ABORTING
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process
TOTAL_APPLIED	NUMBER	Total number of transactions applied by the apply process since the apply process was last started
TOTAL_WAIT_DEPS	NUMBER	Number of times since the apply process was last started that an apply server waited to apply a logical change record (LCR) in a transaction until another apply server applied a transaction because of a dependency between the transactions
TOTAL_WAIT_COMMITS	NUMBER	Number of times since the apply process was last started that an apply server waited to commit a transaction until another apply server committed a transaction to serialize commits

Column	Datatype	Description
TOTAL_ADMIN	NUMBER	Number of administrative jobs issued since the apply process was last started
TOTAL_ASSIGNED	NUMBER	Number of transactions assigned to apply servers since the apply process was last started
TOTAL_RECEIVED	NUMBER	Total number of transactions received by the coordinator process since the apply process was last started
TOTAL_IGNORED	NUMBER	Number of transactions which were received by the coordinator but were ignored because they had been previously applied
TOTAL_ROLLBACKS	NUMBER	Number of transactions which were rolled back due to unexpected contention
TOTAL_ERRORS	NUMBER	Number of transactions applied by the apply process that resulted in an apply error since the apply process was last started
LWM_TIME	DATE	Time when the message with the lowest message number was recorded. The creation time of the message with the lowest message number was also recorded at this time.
LWM_MESSAGE_NUMBER	NUMBER	Number of the message corresponding to the low-watermark. That is, messages with a commit message number less than or equal to this message number have definitely been applied, but some messages with a higher commit message number also may have been applied.
LWM_MESSAGE_CREATE_TIME	DATE	For captured messages, creation time at the source database of the message corresponding to the low-watermark. For user-enqueued messages, time when the message corresponding to the low-watermark was enqueued into the queue at the local database.
HWM_TIME	DATE	Time when the message with the highest message number was recorded. The creation time of the message with the highest message number was also recorded at this time.
HWM_MESSAGE_NUMBER	NUMBER	Number of the message corresponding to the high-watermark. That is, no messages with a commit message number greater than this message number have been applied.
HWM_MESSAGE_CREATE_TIME	DATE	For captured messages, creation time at the source database of the message corresponding to the high-watermark. For user-enqueued messages, time when the message corresponding to the high-watermark was enqueued into the queue at the local database.
STARTUP_TIME	DATE	Time when the apply process was last started
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed (in hundredths of a second) scheduling messages since the apply process was last started
ELAPSED_IDLE_TIME	NUMBER	Elapsed idle time

Note: The ELAPSED_SCHEDULE_TIME column is only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V$STREAMS_APPLY_READER

V$STREAMS_APPLY_READER displays information about each apply reader. The apply reader for an apply process is a process which reads (dequeues) messages from the queue, computes message dependencies, builds transactions, and passes the transactions on to the apply process coordinator in commit order for assignment to the apply servers.

Column	Datatype	Description
SID	NUMBER	Session ID of the reader's session

Column	Datatype	Description
SERIAL#	NUMBER	Serial number of the reader's session
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process that spilled one or more transactions
STATE	VARCHAR2(17)	State of the reader: ■ INITIALIZING - Starting up ■ IDLE - Performing no work ■ DEQUEUE MESSAGES - Dequeueing messages from the apply process queue ■ SCHEDULE MESSAGES - Computing dependencies between messages and assembling messages into transactions ■ SPILLING - Spilling unapplied messages from memory to hard disk ■ PAUSED - Waiting for a DDL LCR to be applied
TOTAL_MESSAGES_ DEQUEUED	NUMBER	Total number of messages dequeued since the apply process was last started
TOTAL_MESSAGES_ SPILLED	NUMBER	Number of messages spilled by the reader since the apply process was last started
DEQUEUE_TIME	DATE	Time when the last message was received
DEQUEUED_MESSAGE_ NUMBER	NUMBER	Number of the last message received
DEQUEUED_MESSAGE_ CREATE_TIME	DATE	For captured messages, creation time at the source database of the last message received. For user-enqueued messages, time when the message was enqueued into the queue at the local database.
SGA_USED	NUMBER	Amount (in bytes) of SGA memory used by the apply process since it was last started
ELAPSED_DEQUEUE_TIME	NUMBER	Time elapsed (in hundredths of a second) dequeuing messages since the apply process was last started
ELAPSED_SCHEDULE_TIME	NUMBER	Time elapsed (in hundredths of a second) scheduling messages since the apply process was last started. Scheduling includes computing dependencies between messages and assembling messages into transactions.
ELAPSED_SPILL_TIME	NUMBER	Elapsed time spent spilling messages (in hundredths of a second) since the apply process was last started
LAST_BROWSE_NUM	NUMBER	Last browse SCN
OLDEST_SCN_NUM	NUMBER	Oldest SCN
LAST_BROWSE_SEQ	NUMBER	Last browse sequence number
LAST_DEQ_SEQ	NUMBER	Last dequeue sequence number
OLDEST_XIDUSN	NUMBER	Transaction ID undo segment number of the oldest transaction to be applied/being applied
OLDEST_XIDSLT	NUMBER	Transaction ID slot number of the oldest transaction to be applied/being applied
OLDEST_XIDSQN	NUMBER	Transaction ID sequence number of the oldest transaction to be applied/ being applied
SPILL_LWM_SCN	NUMBER	Spill low-watermark SCN
PROXY_SID	NUMBER	When the apply process uses combined capture and apply, the session ID of the apply process network receiver that is responsible for direct communication between capture and apply. If the apply process does not use combined capture and apply, then this column is 0.
PROXY_SERIAL	NUMBER	When the apply process uses combined capture and apply, the serial number of the apply process network receiver that is responsible for direct communication between capture and apply. If the apply process does not use combined capture and apply, then this column is 0.

Column	Datatype	Description
PROXY_SPID	VARCHAR2(12)	When the apply process uses combined capture and apply, the process identification number of the apply process network receiver that is responsible for direct communication between capture and apply. If the apply process does not use combined capture and apply, then this column is 0.
CAPTURE_BYTES_RECEIVED	NUMBER	When the apply process uses combined capture and apply, the number of bytes received by the apply process from the capture process since the apply process last started. If the apply process does not use combined capture and apply, then this column is not populated.

Note: The ELAPSED_DEQUEUE_TIME and ELAPSED_SCHEDULE_TIME columns are only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V$STREAMS_APPLY_SERVER

V$STREAMS_APPLY_SERVER displays information about each apply server and its activities. An apply server receives events from the apply coordinator for an apply process. For each event received, an apply server either applies the event or sends the event to the appropriate apply handler.

Column	Datatype	Description
SID	NUMBER	Session ID of the apply server's session
SERIAL#	NUMBER	Serial number of the apply server's session
APPLY#	NUMBER	Apply process number. An apply process is an Oracle background process, prefixed by ap.
APPLY_NAME	VARCHAR2(30)	Name of the apply process
SERVER_ID	NUMBER	Parallel execution server number of the apply server
STATE	VARCHAR2(20)	State of the apply server: ■ IDLE - Performing no work ■ RECORD LOW-WATERMARK - Performing an administrative job that maintains information about the apply progress, which is used in the ALL_APPLY_PROGRESS and DBA_APPLY_PROGRESS data dictionary views ■ ADD PARTITION - Performing an administrative job that adds a partition that is used for recording information about in-progress transactions ■ DROP PARTITION - Performing an administrative job that drops a partition that was used to record information about in-progress transactions ■ EXECUTE TRANSACTION - Applying a transaction ■ WAIT COMMIT - Waiting to commit a transaction until all other transactions with a lower commit SCN are applied. This state is possible only if the COMMIT_SERIALIZATION apply process parameter is set to a value other than none and the PARALLELISM apply process parameter is set to a value greater than 1. ■ WAIT DEPENDENCY - Waiting to apply a logical change record (LCR) in a transaction until another transaction, on which it has a dependency, is applied. This state is possible only if the PARALLELISM apply process parameter is set to a value greater than 1. ■ WAIT FOR NEXT CHUNK - Waiting for the next set of logical change records (LCRs) for a large transaction

Column	Datatype	Description
XIDUSN	NUMBER	Transaction ID undo segment number of the transaction currently being applied
XIDSLT	NUMBER	Transaction ID slot number of the transaction currently being applied
XIDSQN	NUMBER	Transaction ID sequence number of the transaction currently being applied
COMMITSCN	NUMBER	Commit system change number (SCN) of the transaction currently being applied
DEP_XIDUSN	NUMBER	Transaction ID undo segment number of a transaction on which the transaction being applied by this apply server depends
DEP_XIDSLT	NUMBER	Transaction ID slot number of a transaction on which the transaction being applied by this apply server depends
DEP_XIDSQN	NUMBER	Transaction ID sequence number of a transaction on which the transaction being applied by this apply server depends
DEP_COMMITSCN	NUMBER	Commit system change number (SCN) of the transaction on which this apply server depends
MESSAGE_SEQUENCE	NUMBER	Number of the current message being applied by the apply server. This value is reset to 1 at the beginning of each transaction.
TOTAL_ASSIGNED	NUMBER	Total number of transactions assigned to the apply server since the apply process was last started
TOTAL_ADMIN	NUMBER	Total number of administrative jobs done by the apply server since the apply process was last started. See the STATE information in this view for the types of administrative jobs.
TOTAL_ROLLBACKS	NUMBER	Number of transactions assigned to this server which were rolled back
TOTAL_MESSAGES_APPLIED	NUMBER	Total number of messages applied by this apply server since the apply process was last started
APPLY_TIME	DATE	Time the last message was applied
APPLIED_MESSAGE_NUMBER	NUMBER	Number of the last message applied
APPLIED_MESSAGE_CREATE_TIME	DATE	Creation time at the source database of the last captured message applied. No information about user-enqueued messages is recorded in this column.
ELAPSED_DEQUEUE_TIME	NUMBER	Time elapsed (in hundredths of a second) dequeuing messages since the apply process was last started
ELAPSED_APPLY_TIME	NUMBER	Time elapsed (in hundredths of a second) applying messages since the apply process was last started

Note: The ELAPSED_DEQUEUE_TIME and ELAPSED_APPLY_TIME columns are only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V$STREAMS_CAPTURE

V$STREAMS_CAPTURE displays information about each capture process.

Column	Datatype	Description
SID	NUMBER	Session identifier of the capture process
SERIAL#	NUMBER	Session serial number of the capture process session
CAPTURE#	NUMBER	Capture process number. A capture process is an Oracle background process, prefixed by cp.
CAPTURE_NAME	VARCHAR2(30)	Name of the capture process

Column	Datatype	Description
LOGMINER_ID	NUMBER	Session ID of the LogMiner session associated with the capture process
STARTUP_TIME	DATE	Time when the capture process was last started
STATE	VARCHAR2(551)	State of the capture process: ■ INITIALIZING ■ CAPTURING CHANGES ■ EVALUATING RULE ■ ENQUEUING MESSAGE ■ SHUTTING DOWN ■ ABORTING ■ CREATING LCR ■ WAITING FOR DICTIONARY REDO ■ WAITING FOR REDO ■ PAUSED FOR FLOW CONTROL ■ DICTIONARY INITIALIZATION ■ WAITING FOR APPLY TO START ■ CONNECTING TO APPLY DATABASE ■ WAITING FOR PROPAGATION TO START
TOTAL_PREFILTER_ DISCARDED	NUMBER	Total number of prefiltered messages discarded
TOTAL_PREFILTER_KEPT	NUMBER	Total number of prefiltered messages kept
TOTAL_PREFILTER_ EVALUATIONS	NUMBER	Total number of prefilter evaluations
TOTAL_MESSAGES_ CAPTURED	NUMBER	Total number of redo entries passed by LogMiner to the capture process for detailed rule evaluation since the capture process last started. A capture process converts a redo entry into a message and performs detailed rule evaluation on the message when capture process prefiltering cannot discard the change.
CAPTURE_TIME	DATE	Time when the most recent message was captured
CAPTURE_MESSAGE_ NUMBER	NUMBER	Number of the most recently captured message
CAPTURE_MESSAGE_ CREATE_TIME	DATE	Creation time of the most recently captured message
TOTAL_MESSAGES_ CREATED	NUMBER	Count associated with ELAPSED_LCR_TIME to calculate rate
TOTAL_FULL_ EVALUATIONS	NUMBER	Count associated with ELAPSED_RULE_TIME to calculate rate
TOTAL_MESSAGES_ ENQUEUED	NUMBER	Total number of messages enqueued since the capture process was last started
ENQUEUE_TIME	DATE	Time when the last message was enqueued
ENQUEUE_MESSAGE_ NUMBER	NUMBER	Number of the last enqueued message
ENQUEUE_MESSAGE_ CREATE_TIME	DATE	Creation time of the last enqueued message
AVAILABLE_MESSAGE_ NUMBER	NUMBER	For local capture, the last redo SCN flushed to the logfiles. For downstream capture, the last SCN added to LogMiner via the archive logs.
AVAILABLE_MESSAGE_ CREATE_TIME	DATE	For local capture, the time the SCN was written to the logfile. For downstream capture, the time the most recent archive log (containing the most recent SCN) was added to LogMiner.
ELAPSED_CAPTURE_TIME	NUMBER	Elapsed time (in hundredths of a second) scanning for changes in the redo log since the capture process was last started

Column	Datatype	Description
ELAPSED_RULE_TIME	NUMBER	Elapsed time (in hundredths of a second) evaluating rules since the capture process was last started
ELAPSED_ENQUEUE_TIME	NUMBER	Elapsed time (in hundredths of a second) enqueuing messages since the capture process was last started
ELAPSED_LCR_TIME	NUMBER	Elapsed time (in hundredths of a second) creating logical change records (LCRs) since the capture process was last started
ELAPSED_REDO_WAIT_TIME	NUMBER	Elapsed time (in hundredths of a second) spent by the capture process in the WAITING FOR REDO state
ELAPSED_PAUSE_TIME	NUMBER	Elapsed flow control pause time (in hundredths of a second)
STATE_CHANGED_TIME	DATE	Time at which the state of the capture process changed
APPLY_NAME	VARCHAR2(30)	When the capture process uses combined capture and apply, the name of the apply process to which the capture process sends LCRs. If the capture process does not use combined capture and apply, then this column is not populated.
APPLY_DBLINK	VARCHAR2(128)	When the capture process uses combined capture and apply, the database link to the remote database if the apply process is at a remote database. If the capture process does not use combined capture and apply, then this column is not populated.
APPLY_MESSAGES_SENT	NUMBER	When the capture process uses combined capture and apply, the number of messages sent from the capture process to the apply process since the capture process last started. If the capture process does not use combined capture and apply, then this column is not populated.
APPLY_BYTES_SENT	NUMBER	When the capture process uses combined capture and apply, the number of bytes sent from the capture process to the apply process since the capture process last started. If the capture process does not use combined capture and apply, then this column is not populated.

Note: The ELAPSED_CAPTURE_TIME, ELAPSED_RULE_TIME, ELAPSED_ENQUEUE_TIME, ELAPSED_LCR_TIME, and ELAPSED_REDO_WAIT_TIME columns are only populated if the TIMED_STATISTICS initialization parameter is set to true, or if the STATISTICS_LEVEL initialization parameter is set to TYPICAL or ALL.

V$STREAMS_MESSAGE_TRACKING

V$STREAMS_MESSAGE_TRACKING displays information about LCRs tracked through the stream that are processed by each Streams client. Use the DBMS_STREAMS_ADM.SET_MESSAGE_TRACKING procedure to specify a tracking label that becomes part of each LCR generated by the current session.

If the actions parameter in the DBMS_STREAMS_ADM.SET_MESSAGE_TRACKING procedure is set to DBMS_STREAMS_ADM.ACTION_MEMORY, then information about the LCRs is tracked in memory and this view is populated with information about the LCRs. If the actions parameter is set to DBMS_STREAMS_ADM.ACTION_TRACE, then this view is not populated and information about the LCRs is sent to the trace files at each database.

Column	Datatype	Description
TRACKING_LABEL	VARCHAR2(30)	User-specified tracking label
TAG	RAW(30)	First 30 bytes of the tag of the LCR
COMPONENT_NAME	VARCHAR2(30)	Name of the component that processed the LCR
COMPONENT_TYPE	VARCHAR2(30)	Type of the component that processed the LCR

Column	Datatype	Description
ACTION	VARCHAR2(50)	Action performed on the LCR
ACTION_DETAILS	VARCHAR2(100)	Details of the action
TIMESTAMP	TIMESTAMP(13) WITH TIME ZONE	Time when the action was performed
MESSAGE_CREATION_TIME	DATE	Time when the message was created
MESSAGE_NUMBER	NUMBER	SCN of the message
TRACKING_ID	RAW(16)	Globally unique OID of the LCR
SOURCE_DATABASE_NAME	VARCHAR2(128)	Name of the source database
OBJECT_OWNER	VARCHAR2(30)	Owner of the object
OBJECT_NAME	VARCHAR2(30)	Name of the object
XID	VARCHAR2(22)	Transaction ID
COMMAND_TYPE	VARCHAR2(30)	Command type of the LCR

V$STREAMS_POOL_ADVICE

V$STREAMS_POOL_ADVICE displays information about the estimated count of spilled or unspilled messages and the associated time spent in the spill or unspill activity for different Streams pool sizes. The sizes range from 10% to 200% of the current Streams pool size, in equal intervals. The value of the interval depends on the current size of the Streams pool.

Column	Datatype	Description
STREAMS_POOL_SIZE_FOR_ESTIMATE	NUMBER	Sstreams pool size (in megabytes) for the estimate. The size ranges from values smaller than the current Streams pool size to values larger than the current Streams pool size, and there is a separate row for each increment. There is always an entry that shows the current Streams pool size, and there are always 20 increments. The range and the size of the increments depend on the current size of the Streams pool.
STREAMS_POOL_SIZE_FACTOR	NUMBER	Size factor with respect to the current Streams pool size
ESTD_SPILL_COUNT	NUMBER	Estimated count of messages spilled from the Streams pool
ESTD_SPILL_TIME	NUMBER	Estimated elapsed time (in seconds) to spill
ESTD_UNSPILL_COUNT	NUMBER	Estimated count of unspills (read back from disk)
ESTD_UNSPILL_TIME	NUMBER	Estimated elapsed time (in seconds) to unspill

See Also: *Oracle Streams Replication Administrator's Guide* for more information about this view

V$STREAMS_TRANSACTION

V$STREAMS_TRANSACTION displays information about transactions that are being processed by capture processes or apply processes. This view can be used to identify long running transactions and to determine how many logical change records (LCRs) are being processed in each transaction. This view only contains information about captured LCRs. It does not contain information about user-enqueued LCRs or user messages.

This view only shows information about LCRs that are being processed because they satisfied the rule sets for the Streams process at the time of the query. For capture processes, this view only shows information about changes in transactions that the capture process has converted to LCRs. It does not show information about all the active transactions present in the redo log. For apply processes, this view only shows information about LCRs that the apply process has dequeued. It does not show information about LCRs in the apply process's queue.

Information about a transaction remains in the view until the transaction commits or until the entire transaction is rolled back.

Column	Datatype	Description
STREAMS_NAME	VARCHAR2(30)	Name of the Streams process
STREAMS_TYPE	VARCHAR2(10)	Type of the Streams process: ■ CAPTURE ■ APPLY
XIDUSN	NUMBER	Transaction ID undo segment number of the transaction
XIDSLT	NUMBER	Transaction ID slot number of the transaction
XIDSQN	NUMBER	Transaction ID sequence number of the transaction
CUMULATIVE_MESSAGE_COUNT	NUMBER	Number of LCRs processed in the transaction. If the Streams process is restarted while the transaction is being processed, then this column shows the number of LCRs processed in the transaction since the Streams process was started.
TOTAL_MESSAGE_COUNT	NUMBER	Total Number of LCRs processed in the transaction by an apply process. This column does not pertain to capture processes.
FIRST_MESSAGE_TIME	DATE	Timestamp of the first LCR processed in the transaction. If a capture process is restarted while the transaction is being processed, then this column shows the timestamp of the first LCR processed after the capture process was started.
FIRST_MESSAGE_NUMBER	NUMBER	System change number (SCN) of the first message in the transaction. If a capture process is restarted while the transaction is being processed, then this column shows the SCN of the first message processed after the capture process was started.
LAST_MESSAGE_TIME	DATE	Timestamp of the last LCR processed in the transaction
LAST_MESSAGE_NUMBER	NUMBER	SCN of the most recent message encountered for the transaction

V$SUBCACHE

V$SUBCACHE displays information about the subordinate caches currently loaded into library cache memory. The view walks through the library cache, printing out a row for each loaded subordinate cache per library cache object.

Column	Datatype	Description
OWNER_NAME	VARCHAR2(64)	Owner of the object containing these cache entries
NAME	VARCHAR2(1000)	Object Name
TYPE	NUMBER	Object Type
HEAP_NUM	NUMBER	Heap number containing this subordinate cache
CACHE_ID	NUMBER	Subordinate cache ID
CACHE_CNT	NUMBER	Number of entries for this cache in this object
HEAP_SZ	NUMBER	Amount of extent space allocated to this heap
HEAP_ALOC	NUMBER	Amount of extent space allocated from this heap
HEAP_USED	NUMBER	Amount of space utilized in this heap

V$SUBSCR_REGISTRATION_STATS

V$SUBSCR_REGISTRATION_STATS displays information for diagnosability of notifications.

Column	Datatype	Description
REG_ID	NUMBER	Registration identifier
NUM_NTFNS	NUMBER	Number of notifications
NUM_GROUPING_NTFNS	NUMBER	Number of grouping notifications
LAST_NTFN_START_TIME	TIMESTAMP(3) WITH TIME ZONE	Time when the last notification was started
LAST_NTFN_SENT_TIME	TIMESTAMP(3) WITH TIME ZONE	Time when the last notification was sent
TOTAL_EMON_LATENCY	NUMBER	Total EMON latency (time taken by the EMON slave to process notifications)
EMON#	NUMBER	Active EMON slave serving the registration
ALL_EMON_SERVERS	RAW(2000)	EMON slaves that served the registration
TOTAL_PAYLOAD_BYTES_SENT	NUMBER	Total payload bytes sent
NUM_RETRIES	NUMBER	Number of retries in sending notifications
TOTAL_PLSQL_EXEC_TIME	NUMBER	Total PL/SQL callback execution time (relevant only for PL/SQL notifications)
LAST_ERR	VARCHAR2(90)	Last error message
LAST_ERR_TIME	TIMESTAMP(3) WITH TIME ZONE	Time of the last error
LAST_UPDATE_TIME	TIMESTAMP(3) WITH TIME ZONE	Time of the last update

V$SYS_OPTIMIZER_ENV

V$SYS_OPTIMIZER_ENV displays the contents of the optimizer environment for the instance. The optimizer environment stores the value of the main parameters used by the Oracle optimizer when building the execution plan of a SQL statement. Hence, modifying the value of one or more of these parameters (for example, by issuing an ALTER SYSTEM statement) could lead to plan changes.

The parameters displayed by this view are either regular initialization parameters (such as OPTIMIZER_FEATURES_ENABLE) or pseudo parameters (such as ACTIVE_INSTANCE_COUNT).

Column	Datatype	Description
ID	NUMBER	Unique identifier of the parameter in the optimizer environment
NAME	VARCHAR2(40)	Name of the parameter
SQL_FEATURE	VARCHAR2(64)	Associated feature control ID
ISDEFAULT	VARCHAR2(3)	Indicates whether the parameter is set to the default value (YES) or not (NO)
VALUE	VARCHAR2(25)	Value of the parameter
DEFAULT_VALUE	VARCHAR2(25)	Default value of the parameter

V$SYS_TIME_MODEL

V$SYS_TIME_MODEL displays the system-wide accumulated times for various operations. The time reported is the total elapsed or CPU time (in microseconds). Any timed operation will buffer at most 5 seconds of time data. Specifically, this means that if a timed operation (such as SQL execution) takes a long period of time to perform, the data published to this view is at most missing 5 seconds of the time accumulated for the operation.

The time values are 8-byte integers and can therefore hold approximately 580,000 years worth of time before wrapping. Background process time is not included in a statistic value unless the statistic is specifically for background processes.

Column	Datatype	Description
STAT_ID	NUMBER	Statistic identifier for the time statistic
STAT_NAME	VARCHAR2(64)	Name of the statistic (see Table 8–1 on page 8-7)
VALUE	NUMBER	Amount of time (in microseconds) that the system has spent in this operation

V$SYSAUX_OCCUPANTS

V$SYSAUX_OCCUPANTS displays SYSAUX tablespace occupant information.

Column	Datatype	Description
OCCUPANT_NAME	VARCHAR2(64)	Occupant name
OCCUPANT_DESC	VARCHAR2(64)	Occupant description
SCHEMA_NAME	VARCHAR2(64)	Schema name for the occupant
MOVE_PROCEDURE	VARCHAR2(64)	Name of the move procedure; null if not applicable
MOVE_PROCEDURE_DESC	VARCHAR2(64)	Description of the move procedure
SPACE_USAGE_KBYTES	NUMBER	Current space usage of the occupant (in KB)

V$SYSMETRIC

V$SYSMETRIC displays the system metric values captured for the most current time interval for both the long duration (60-second) and short duration (15-second) system metrics.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric name
VALUE	NUMBER	Metric value
METRIC_UNIT	VARCHAR2(64)	Metric unit description

V$SYSMETRIC_HISTORY

V$SYSMETRIC_HISTORY displays all system metric values available in the database. Both long duration (60-second with 1 hour history) and short duration (15-second with one-interval only) metrics are displayed by this view.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric name
VALUE	NUMBER	Metric value
METRIC_UNIT	VARCHAR2(64)	Metric unit description

V$SYSMETRIC_SUMMARY

V$SYSMETRIC_SUMMARY displays a summary of all system Metric values for the long-duration system metrics. The average, maximum value, minimum value, and the value of one standard deviation for the last hour are displayed for each metric item.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
GROUP_ID	NUMBER	Metric group ID
METRIC_ID	NUMBER	Metric ID
METRIC_NAME	VARCHAR2(64)	Metric name
NUM_INTERVAL	NUMBER	Number of intervals observed
MAXVAL	NUMBER	Maximum value observed
MINVAL	NUMBER	Minimum value observed
AVERAGE	NUMBER	Average value over the period
STANDARD_DEVIATION	NUMBER	One standard deviation
METRIC_UNIT	VARCHAR2(64)	Metric unit description

V$SYSSTAT

V$SYSSTAT displays system statistics. To find the name of the statistic associated with each statistic number (STATISTIC#), query the V$STATNAME view.

Column	Datatype	Description
STATISTIC#	NUMBER	Statistic number **Note:** Statistics numbers are not guaranteed to remain constant from one release to another. Therefore, you should rely on the statistics name rather than its number in your applications.
NAME	VARCHAR2(64)	Statistic name. You can get a complete listing of statistic names by querying the V$STATNAME view.

Column	Datatype	Description
CLASS	NUMBER	A number representing one or more statistics class. The following class numbers are additive: ■ 1 - User ■ 2 - Redo ■ 4 - Enqueue ■ 8 - Cache ■ 16 - OS ■ 32 - Real Application Clusters ■ 64 - SQL ■ 128 - Debug
VALUE	NUMBER	Statistic value
STAT_ID	NUMBER	Identifier of the statistic

See Also: "V$STATNAME" on page 8-55 and Appendix E, "Statistics Descriptions"

V$SYSTEM_CURSOR_CACHE

V$SYSTEM_CURSOR_CACHE displays system wide information on cursor usage.

See Also: "V$SESSION_CURSOR_CACHE" on page 8-15

Column	Datatype	Description
OPENS	NUMBER	Cumulative total of cursor opens
HITS	NUMBER	Cumulative total of cursor open hits
HIT_RATIO	NUMBER	Ratio of the number of times an open cursor was found divided by the number of times a cursor was sought

V$SYSTEM_EVENT

V$SYSTEM_EVENT displays information on total waits for an event. Note that the TIME_WAITED and AVERAGE_WAIT columns will contain a value of zero on those platforms that do not support a fast timing mechanism. If you are running on one of these platforms and you want this column to reflect true wait times, you must set TIMED_STATISTICS to TRUE in the parameter file; doing this will have a small negative effect on system performance.

See Also: "TIMED_STATISTICS" on page 1-160

Column	Datatype	Description
EVENT	VARCHAR2(64)	Name of the wait event
TOTAL_WAITS	NUMBER	Total number of waits for the event
TOTAL_TIMEOUTS	NUMBER	Total number of timeouts for the event
TIME_WAITED	NUMBER	Total amount of time waited for the event (in hundredths of a second)
AVERAGE_WAIT	NUMBER	Average amount of time waited for the event (in hundredths of a second)
TIME_WAITED_MICRO	NUMBER	Total amount of time waited for the event (in microseconds)
TOTAL_WAITS_FG	NUMBER	Total number of waits for the event, from foreground sessions
TOTAL_TIMEOUTS_FG	NUMBER	Total number of timeouts for the event, from foreground sessions

Column	Datatype	Description
TIME_WAITED_FG	NUMBER	Amount of time waited for the event (in hundredths of a second), from foreground sessions
AVERAGE_WAIT_FG	NUMBER	Average amount of time waited for the event (in hundredths of a second), from foreground sessions
TIME_WAITED_MICRO_FG	NUMBER	Amount of time waited for the event (in microseconds), from foreground sessions
EVENT_ID	NUMBER	Identifier of the wait event
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS	VARCHAR2(64)	Name of the class of the wait event

V$SYSTEM_FIX_CONTROL

V$SYSTEM_FIX_CONTROL displays information about Fix Control (enabled/disabled) at the system level.

Column	Datatype	Description
BUGNO	NUMBER	Bug number (as fix control identifier)
VALUE	NUMBER	Current value set for the fix control
SQL_FEATURE	VARCHAR2(64)	Feature control ID
DESCRIPTION	VARCHAR2(64)	Description of the fix control
OPTIMIZER_FEATURE_ENABLE	VARCHAR2(25)	Version on (and after) which the fix is enabled by default
EVENT	NUMBER	Event formerly used to control this fix
IS_DEFAULT	NUMBER	Indicates whether the current value is the same as the default (1) or not (0)

V$SYSTEM_PARAMETER

V$SYSTEM_PARAMETER displays information about the initialization parameters that are currently in effect for the instance. A new session inherits parameter values from the instance-wide values.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter
TYPE	NUMBER	Parameter type: ■ 1 - Boolean ■ 2 - String ■ 3 - Integer ■ 4 - Parameter file ■ 5 - Reserved ■ 6 - Big integer
VALUE	VARCHAR2(4000)	Instance-wide parameter value
DISPLAY_VALUE	VARCHAR2(4000)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(9)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)

Column	Datatype	Description
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: ■ IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. ■ DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. ■ FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEm, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SYSTEM was performed, the value will be MODIFIED.
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
ISBASIC	VARCHAR2(5)	Indicates whether the parameter is a basic parameter (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update
HASH	NUMBER	Hash value for the parameter name

V$SYSTEM_PARAMETER2

V$SYSTEM_PARAMETER2 displays information about the initialization parameters that are currently in effect for the instance, with each list parameter value appearing as a row in the view. A new session inherits parameter values from the instance-wide values.

Presenting the list parameter values in this format enables you to quickly determine the values for a list parameter. For example, if a parameter value is a, b, then the V$SYSTEM_PARAMETER view does not tell you if the parameter has two values (both a and b) or one value (a, b). V$SYSTEM_PARAMETER2 makes the distinction between the list parameter values clear.

Column	Datatype	Description
NUM	NUMBER	Parameter number
NAME	VARCHAR2(80)	Name of the parameter

Column	Datatype	Description
TYPE	NUMBER	Parameter type: - 1 - Boolean - 2 - String - 3 - Integer - 4 - Parameter file - 5 - Reserved - 6 - Big integer
VALUE	VARCHAR2(4000)	Parameter value
DISPLAY_VALUE	VARCHAR2(4000)	Parameter value in a user-friendly format. For example, if the VALUE column shows the value 262144 for a big integer parameter, then the DISPLAY_VALUE column will show the value 256K.
ISDEFAULT	VARCHAR2(6)	Indicates whether the parameter is set to the default value (TRUE) or the parameter value was specified in the parameter file (FALSE)
ISSES_MODIFIABLE	VARCHAR2(5)	Indicates whether the parameter can be changed with ALTER SESSION (TRUE) or not (FALSE)
ISSYS_MODIFIABLE	VARCHAR2(9)	Indicates whether the parameter can be changed with ALTER SYSTEM and when the change takes effect: - IMMEDIATE - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect immediately. - DEFERRED - Parameter can be changed with ALTER SYSTEM regardless of the type of parameter file used to start the instance. The change takes effect in subsequent sessions. - FALSE - Parameter cannot be changed with ALTER SYSTEM unless a server parameter file was used to start the instance. The change takes effect in subsequent instances.
ISINSTANCE_MODIFIABLE	VARCHAR2(5)	For parameters that can be changed with ALTER SYSTEm, indicates whether the value of the parameter can be different for every instance (TRUE) or whether the parameter must have the same value for all Real Application Clusters instances (FALSE). If the ISSYS_MODIFIABLE column is FALSE, then this column is always FALSE.
ISMODIFIED	VARCHAR2(8)	Indicates how the parameter was modified. If an ALTER SYSTEM was performed, the value will be MODIFIED.
ISADJUSTED	VARCHAR2(5)	Indicates whether Oracle adjusted the input value to a more suitable value (for example, the parameter value should be prime, but the user input a non-prime number, so Oracle adjusted the value to the next prime number)
ISDEPRECATED	VARCHAR2(5)	Indicates whether the parameter has been deprecated (TRUE) or not (FALSE)
ISBASIC	VARCHAR2(5)	Indicates whether the parameter is a basic parameter (TRUE) or not (FALSE)
DESCRIPTION	VARCHAR2(255)	Description of the parameter
ORDINAL	NUMBER	Position (ordinal number) of the parameter value. Useful only for parameters whose values are lists of strings.
UPDATE_COMMENT	VARCHAR2(255)	Comments associated with the most recent update

V$SYSTEM_WAIT_CLASS

V$SYSTEM_WAIT_CLASS displays the instance-wide time totals for each registered wait class.

Column	Datatype	Description
WAIT_CLASS_ID	NUMBER	Identifier of the wait class
WAIT_CLASS#	NUMBER	Number of the wait class

Column	Datatype	Description
WAIT_CLASS	VARCHAR2(64)	Name of the wait class
TOTAL_WAITS	NUMBER	Number of times waits of the class occurred
TIME_WAITED	NUMBER	Amount of time spent in the wait by all sessions in the instance
TOTAL_WAITS_FG	NUMBER	Number of times waits of the class occurred, in the foreground session
TIME_WAITED_FG	NUMBER	Amount of time spent in the wait, in the foreground session

V$TABLESPACE

V$TABLESPACE displays tablespace information from the control file.

Column	Datatype	Description
TS#	NUMBER	Tablespace number
NAME	VARCHAR2(30)	Tablespace name
INCLUDED_IN_DATABASE_BACKUP	VARCHAR2(3)	Indicates whether the tablespace is included in full database backups using the BACKUP DATABASE RMAN command (YES) or not (NO); NO only if the CONFIGURE EXCLUDE RMAN command was used for this tablespace
BIGFILE	VARCHAR2(3)	Indicates whether the tablespace is a bigfile tablespace (YES) or a smallfile tablespace (NO)
FLASHBACK_ON	VARCHAR2(3)	Indicates whether the tablespace participates in FLASHBACK DATABASE operations (YES) or not (NO)
ENCRYPT_IN_BACKUP	VARCHAR2(3)	Indicates whether encryption is turned ON or off at the tablespace level: ■ ON - Encryption is turned ON at the tablespace level ■ OFF - Encryption is turned OFF at the tablespace level ■ NULL - Encryption is neither explicitly turned on nor off at the tablespace level (default or when cleared)

V$TEMP_CACHE_TRANSFER

V$TEMP_CACHE_TRANSFER is deprecated. The information that was provided in this view is now provided in the V$INSTANCE_CACHE_TRANSFER and V$SEGMENT_STATISTICS views.

Column	Datatype	Description
FILE_NUMBER	NUMBER	Number of the tempfile
X_2_NULL	NUMBER	Number of blocks with Exclusive-to-NULL conversions; always 0
X_2_NULL_FORCED_WRITE	NUMBER	Number of Exclusive-to-NULL forced writes; always 0
X_2_NULL_FORCED_STALE	NUMBER	Number of Exclusive-to-NULL blocks converted to CR; always 0
X_2_S	NUMBER	Number of blocks with Exclusive-to-Shared conversions; always 0
X_2_S_FORCED_WRITE	NUMBER	Number of Exclusive-to-Shared forced writes; always 0
S_2_NULL	NUMBER	Number of blocks with Shared-to-NULL conversions; always 0
S_2_NULL_FORCED_STALE	NUMBER	Number of Shared-to-NULL blocks converted to CR; always 0
RBR	NUMBER	Number of reuse blocks cross-instance calls; always 0
RBR_FORCED_WRITE	NUMBER	Number of blocks written due to reuse blocks cross-instance calls; always 0
NULL_2_X	NUMBER	Number of blocks with NULL-to-Exclusive conversions; always 0
S_2_X	NUMBER	Number of blocks with Shared-to-Exclusive conversions; always 0
NULL_2_S	NUMBER	Number of blocks with NULL-to-Shared conversions; always 0

V$TEMP_EXTENT_MAP

V$TEMP_EXTENT_MAP displays the status of each unit for all LOCALLY MANAGED temporary tablespaces.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace this unit belongs to
FILE_ID	NUMBER	Absolute file number
BLOCK_ID	NUMBER	Begin block number for this unit
BYTES	NUMBER	Bytes in the extent
BLOCKS	NUMBER	Blocks in the extent
OWNER	NUMBER	Instance which owns this unit
RELATIVE_FNO	NUMBER	Relative file number

V$TEMP_EXTENT_POOL

V$TEMP_EXTENT_POOL displays the state of temporary space cached and used for the instance. Note that loading of the temporary space cache is lazy and that instances can be dormant.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace
FILE_ID	NUMBER	Absolute file number
EXTENTS_CACHED	NUMBER	Number of extents that have been cached
EXTENTS_USED	NUMBER	Number of extents that are actually being used
BLOCKS_CACHED	NUMBER	Number of blocks that are cached
BLOCKS_USED	NUMBER	Number of blocks that are used
BYTES_CACHED	NUMBER	Number of bytes that are cached
BYTES_USED	NUMBER	Number of bytes that are used
RELATIVE_FNO	NUMBER	Relative file number

V$TEMP_SPACE_HEADER

V$TEMP_SPACE_HEADER displays aggregate information per file per LOCALLY MANAGED temporary tablespace regarding how much space is currently being used and how much is free as identified in the space header.

Column	Datatype	Description
TABLESPACE_NAME	VARCHAR2(30)	Name of the temporary tablespace
FILE_ID	NUMBER	Absolute file number
BYTES_USED	NUMBER	How many bytes are in use
BLOCKS_USED	NUMBER	How many blocks are in use
BYTES_FREE	NUMBER	How many bytes are free
BLOCKS_FREE	NUMBER	How many blocks are free
RELATIVE_FNO	NUMBER	The relative file number for the file

V$TEMPFILE

V$TEMPFILE displays tempfile information.

Column	Datatype	Description
FILE#	NUMBER	Absolute file number
CREATION_CHANGE#	NUMBER	Creation System Change Number (SCN)
CREATION_TIME	DATE	Creation time
TS#	NUMBER	Tablespace number
RFILE#	NUMBER	Relative file number in the tablespace
STATUS	VARCHAR2(7)	Status of the file (OFFLINE|ONLINE)
ENABLED	VARCHAR2(10)	Enabled for read and/or write
BYTES	NUMBER	Size of the file in bytes (from the file header)
BLOCKS	NUMBER	Size of the file in blocks (from the file header)
CREATE_BYTES	NUMBER	Creation size of the file (in bytes)
BLOCK_SIZE	NUMBER	Block size for the file
NAME	VARCHAR2(513)	Name of the file

V$TEMPORARY_LOBS

V$TEMPORARY_LOBS displays temporary LOBs.

Column	Datatype	Description
SID	NUMBER	Session ID
CACHE_LOBS	NUMBER	Number of cache temp LOBs
NOCACHE_LOBS	NUMBER	Number of nocache temp LOBs
ABSTRACT_LOBS	NUMBER	Number of abstract LOBs

V$TEMPSEG_USAGE

V$TEMPSEG_USAGE describes temporary segment usage.

Column	Datatype	Description
USERNAME	VARCHAR2(30)	User who requested temporary space
USER	VARCHAR2(30)	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in USERNAME.
SESSION_ADDR	RAW(4 | 8)	Address of shared SQL cursor
SESSION_NUM	NUMBER	Serial number of session
SQLADDR	RAW(4 | 8)	Address of SQL statement
SQLHASH	NUMBER	Hash value of SQL statement
SQL_ID	VARCHAR2(13)	SQL identifier of SQL statement
TABLESPACE	VARCHAR2(31)	Tablespace in which space is allocated
CONTENTS	VARCHAR2(9)	Indicates whether tablespace is TEMPORARY or PERMANENT

Column	Datatype	Description
SEGTYPE	VARCHAR2(9)	Type of sort segment: ■ SORT ■ HASH ■ DATA ■ INDEX ■ LOB_DATA ■ LOB_INDEX
SEGFILE#	NUMBER	File number of initial extent
SEGBLK#	NUMBER	Block number of the initial extent
EXTENTS	NUMBER	Extents allocated to the sort
BLOCKS	NUMBER	Extents in blocks allocated to the sort
SEGRFNO#	NUMBER	Relative file number of initial extent

V$TEMPSTAT

V$TEMPSTAT displays information about file read/write statistics.

Column	Datatype	Description
FILE#	NUMBER	Number of the file
PHYRDS	NUMBER	Number of physical reads done
PHYWRTS	NUMBER	Number of times DBWR is required to write
PHYBLKRD	NUMBER	Number of physical blocks read
PHYBLKWRT	NUMBER	Number of blocks written to disk, which may be the same as PHYWRTS if all writes are single blocks
SINGLEBLKRDS	NUMBER	Number of single block reads
READTIM	NUMBER	Time (in hundredths of a second) spent doing reads if the TIMED_STATISTICS parameter is true; 0 if false
WRITETIM	NUMBER	Time (in hundredths of a second) spent doing writes if the TIMED_STATISTICS parameter is true; 0 if false
SINGLEBLKRDTIM	NUMBER	Cumulative single block read time (in hundredths of a second)
AVGIOTIM	NUMBER	Average time (in hundredths of a second) spent on I/O, if the TIMED_STATISTICS parameter is true; 0 if false
LSTIOTIM	NUMBER	Time (in hundredths of a second) spent doing the last I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MINIOTIM	NUMBER	Minimum time (in hundredths of a second) spent on a single I/O, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIORTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single read, if the TIMED_STATISTICS parameter is true; 0 if false
MAXIOWTM	NUMBER	Maximum time (in hundredths of a second) spent doing a single write, if the TIMED_STATISTICS parameter is true; 0 if false

V$THREAD

V$THREAD displays thread information from the control file.

Column	Datatype	Description
THREAD#	NUMBER	Thread number
STATUS	VARCHAR2(6)	Thread status (OPEN | CLOSED)

Column	Datatype	Description
ENABLED	VARCHAR2(8)	Enabled status: DISABLED, (enabled) PRIVATE, or (enabled) PUBLIC
GROUPS	NUMBER	Number of log groups assigned to this thread
INSTANCE	VARCHAR2(16)	Instance name, if available
OPEN_TIME	DATE	Last time the thread was opened
CURRENT_GROUP#	NUMBER	Current log group
SEQUENCE#	NUMBER	Sequence number of current log
CHECKPOINT_CHANGE#	NUMBER	SCN at last checkpoint
CHECKPOINT_TIME	DATE	Time of last checkpoint
ENABLE_CHANGE#	NUMBER	SCN at which thread was enabled
ENABLE_TIME	DATE	Time of enable SCN
DISABLE_CHANGE#	NUMBER	SCN at which thread was disabled
DISABLE_TIME	DATE	Time of disable SCN
LAST_REDO_SEQUENCE#	NUMBER	Last redo sequence number written by LGWR
LAST_REDO_BLOCK	NUMBER	Last redo block written by LGWR
LAST_REDO_CHANGE#	NUMBER	SCN of last redo for the thread
LAST_REDO_TIME	DATE	Time of last redo for the thread

V$THRESHOLD_TYPES

V$THRESHOLD_TYPES displays information about threshold types.

Column	Datatype	Description
METRICS_ID	NUMBER	Metrics ID
METRICS_GROUP_ID	NUMBER	Metrics group ID
OPERATOR_MASK	NUMBER	Operator mask
OBJECT_TYPE	VARCHAR2(64)	Object type: ■ SYSTEM ■ FILE ■ SERVICE ■ EVENT_CLASS ■ TABLESPACE ■ SESSION
ALERT_REASON_ID	NUMBER	ID of the alert reason
METRIC_VALUE_TYPE	NUMBER	Metric value type

V$TIMER

V$TIMER displays the elapsed time in hundredths of a second. Time is measured since the beginning of the epoch, which is operating system specific, and wraps around to 0 again whenever the value overflows four bytes (roughly 497 days).

Column	Datatype	Description
HSECS	NUMBER	Elapsed time (in hundredths of a second)

V$TIMEZONE_NAMES

V$TIMEZONE_NAMES displays valid time zone names.

Column	Datatype	Description
TZNAME	VARCHAR2(64)	Time zone region (for example, US/Pacific)
TZABBREV	VARCHAR2(64)	Corresponding daylight abbreviation (for example, PDT)

V$TRANSACTION

V$TRANSACTION lists the active transactions in the system.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of the transaction state object
XIDUSN	NUMBER	Undo segment number
XIDSLOT	NUMBER	Slot number
XIDSQN	NUMBER	Sequence number
UBAFIL	NUMBER	Undo block address (UBA) filenum
UBABLK	NUMBER	UBA block number
UBASQN	NUMBER	UBA sequence number
UBAREC	NUMBER	UBA record number
STATUS	VARCHAR2(16)	Status
START_TIME	VARCHAR2(20)	Start time (wall clock)
START_SCNB	NUMBER	Start system change number (SCN) base
START_SCNW	NUMBER	Start SCN wrap
START_UEXT	NUMBER	Start extent number
START_UBAFIL	NUMBER	Start UBA file number
START_UBABLK	NUMBER	Start UBA block number
START_UBASQN	NUMBER	Start UBA sequence number
START_UBAREC	NUMBER	Start UBA record number
SES_ADDR	RAW(4 | 8)	User session object address
FLAG	NUMBER	Flag
SPACE	VARCHAR2(3)	YES if a space transaction
RECURSIVE	VARCHAR2(3)	YES if a recursive transaction
NOUNDO	VARCHAR2(3)	YES if a no undo transaction
PTX	VARCHAR 2(3)	YES if parallel transaction
NAME	VARCHAR2(256)	Name of a named transaction
PRV_XIDUSN	NUMBER	Previous transaction undo segment number
PRV_XIDSLT	NUMBER	Previous transaction slot number
PRV_XIDSQN	NUMBER	Previous transaction sequence number
PTX_XIDUSN	NUMBER	Rollback segment number of the parent XID
PTX_XIDSLT	NUMBER	Slot number of the parent XID
PTX_XIDSQN	NUMBER	Sequence number of the parent XID
DSCN-B	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in DSCN_BASE.
DSCN-W	NUMBER	This column is obsolete and maintained for backward compatibility. The value of this column is always equal to the value in DSCN_WRAP.

Column	Datatype	Description
USED_UBLK	NUMBER	Number of undo blocks used
USED_UREC	NUMBER	Number of undo records used
LOG_IO	NUMBER	Logical I/O
PHY_IO	NUMBER	Physical I/O
CR_GET	NUMBER	Consistent gets
CR_CHANGE	NUMBER	Consistent changes
START_DATE	DATE	Start time (wall clock)
DSCN_BASE	NUMBER	Dependent SCN base
DSCN_WRAP	NUMBER	Dependent SCN wrap
START_SCN	NUMBER	Start SCN
DEPENDENT_SCN	NUMBER	Dependent SCN
XID	RAW(8)	Transaction XID
PRV_XID	RAW(8)	Previous transaction XID
PTX_XID	RAW(8)	Parent transaction XID

V$TRANSACTION_ENQUEUE

V$TRANSACTION_ENQUEUE displays locks owned by transaction state objects.

Column	Datatype	Description
ADDR	RAW(4 | 8)	Address of lock state object
KADDR	RAW(4 | 8)	Address of lock
SID	NUMBER	Identifier for session holding or acquiring the lock
TYPE	VARCHAR2(2)	Type of lock. TX indicates transaction enqueue.
ID1	NUMBER	Lock identifier #1 (depends on type)
ID2	NUMBER	Lock identifier #2 (depends on type)
LMODE	NUMBER	Lock mode in which the session holds the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
REQUEST	NUMBER	Lock mode in which the process requests the lock: ■ 0 - none ■ 1 - null (NULL) ■ 2 - row-S (SS) ■ 3 - row-X (SX) ■ 4 - share (S) ■ 5 - S/Row-X (SSX) ■ 6 - exclusive (X)
CTIME	NUMBER	Time since current mode was granted
BLOCK	NUMBER	The lock is blocking another lock

V$TRANSPORTABLE_PLATFORM

V$TRANSPORTABLE_PLATFORM displays all platforms that support cross-platform tablespace transport. Specifically, it lists all platforms supported by the RMAN CONVERT TABLESPACE command, along with the endianness of each platform.

Column	Datatype	Description
PLATFORM_ID	NUMBER	Platform identification number
PLATFORM_NAME	VARCHAR2(101)	Platform name
ENDIAN_FORMAT	VARCHAR2(14)	Platform endian format: ▪ Big ▪ Little ▪ UNKNOWN FORMAT

V$TYPE_SIZE

V$TYPE_SIZE displays the sizes of various database components for use in estimating data block capacity.

Column	Datatype	Description
COMPONENT	VARCHAR2(8)	Component name, such as segment or buffer header
TYPE	VARCHAR2(8)	Component type
DESCRIPTION	VARCHAR2(32)	Description of the component
TYPE_SIZE	NUMBER	Size of the component

V$UNDOSTAT

V$UNDOSTAT displays a histogram of statistical data to show how well the system is working. The available statistics include undo space consumption, transaction concurrency, and length of queries executed in the instance. You can use this view to estimate the amount of undo space required for the current workload. Oracle uses this view to tune undo usage in the system. The view returns NULL values if the system is in manual undo management mode.

Each row in the view keeps statistics collected in the instance for a 10-minute interval. The rows are in descending order by the BEGIN_TIME column value. Each row belongs to the time interval marked by (BEGIN_TIME, END_TIME). Each column represents the data collected for the particular statistic in that time interval. The first row of the view contains statistics for the (partial) current time period. The view contains a total of 576 rows, spanning a 4 day cycle.

Column	Datatype	Description
BEGIN_TIME	DATE	Identifies the beginning of the time interval
END_TIME	DATE	Identifies the end of the time interval
UNDOTSN	NUMBER	Represents the last active undo tablespace in the duration of time. The tablespace ID of the active undo tablespace is returned in this column. If more than one undo tablespace was active in that period, the active undo tablespace that was active at the end of the period is reported.
UNDOBLKS	NUMBER	Represents the total number of undo blocks consumed. You can use this column to obtain the consumption rate of undo blocks, and thereby estimate the size of the undo tablespace needed to handle the workload on your system.
TXNCOUNT	NUMBER	Identifies the total number of transactions executed within the period

Column	Datatype	Description
MAXQUERYLEN	NUMBER	Identifies the length of the longest query (in seconds) executed in the instance during the period. You can use this statistic to estimate the proper setting of the UNDO_RETENTION initialization parameter. The length of a query is measured from the cursor open time to the last fetch/execute time of the cursor. Only the length of those cursors that have been fetched/executed during the period are reflected in the view.
MAXQUERYID	VARCHAR2(13)	SQL identifier of the longest running SQL statement in the period
MAXCONCURRENCY	NUMBER	Identifies the highest number of transactions executed concurrently within the period
UNXPSTEALCNT	NUMBER	Number of attempts to obtain undo space by stealing unexpired extents from other transactions
UNXPBLKRELCNT	NUMBER	Number of unexpired blocks removed from certain undo segments so they can be used by other transactions
UNXPBLKREUCNT	NUMBER	Number of unexpired undo blocks reused by transactions
EXPSTEALCNT	NUMBER	Number of attempts to steal expired undo blocks from other undo segments
EXPBLKRELCNT	NUMBER	Number of expired undo blocks stolen from other undo segments
EXPBLKREUCNT	NUMBER	Number of expired undo blocks reused within the same undo segments
SSOLDERRCNT	NUMBER	Identifies the number of times the error ORA-01555 occurred. You can use this statistic to decide whether or not the UNDO_RETENTION initialization parameter is set properly given the size of the undo tablespace. Increasing the value of UNDO_RETENTION can reduce the occurrence of this error.
NOSPACEERRCNT	NUMBER	Identifies the number of times space was requested in the undo tablespace and there was no free space available. That is, all of the space in the undo tablespace was in use by active transactions. The corrective action is to add more space to the undo tablespace.
ACTIVEBLKS	NUMBER	Total number of blocks in the active extents of the undo tablespace for the instance at the sampled time in the period
UNEXPIREDBLKS	NUMBER	Total number of blocks in the unexpired extents of the undo tablespace for the instance at the sampled time in the period
EXPIREDBLKS	NUMBER	Total number of blocks in the expired extents of the undo tablespace for the instance at the sampled time in the period
TUNED_UNDORETENTION	NUMBER	Amount of time (in seconds) for which undo will not be recycled from the time it was committed. At any point in time, the latest value of TUNED_UNDORETENTION is used to determine whether data committed at a particular time in the past can be recycled.

V$UNUSABLE_BACKUPFILE_DETAILS

V$UNUSABLE_BACKUPFILE_DETAILS displays information about all backup files (backup pieces, proxy copies, or copies) that are marked unavailable and expired. You can select one of the rows and use BTYPE_KEY or FILETYPE_KEY to change the status of a backup file set or a specific file to available.

Column	Datatype	Description
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Session record ID
SESSION_STAMP	NUMBER	Session stamp
RMAN_STATUS_RECID	NUMBER	Record ID of the corresponding row in the control file
RMAN_STATUS_STAMP	NUMBER	Timestamp of the row in the controlfile
BTYPE	CHAR(9)	Backup type container. Possible values are: BACKUPSET, IMAGECOPY, PROXYCOPY.
BTYPE_KEY	NUMBER	Unique identifier for the backup type, either BS_KEY or COPY_KEY.

Column	Datatype	Description
ID1	NUMBER	If BACKUPSET, it contains SET_STAMP. If IMAGECOPY or PROXYCOPY, it is RECID from the control file.
ID2	NUMBER	If BACKUPSET, it contains SET_COUNT. If IMAGECOPY or PROXYCOPY, it is STAMP.
FILETYPE	VARCHAR2(15)	Type of file. Possible values are: BACKUPPIECE, COPY, PROXYCOPY.
FILETYPE_KEY	NUMBER	Backup piece key if the file is a backup piece; otherwise COPY_KEY.
STATUS	VARCHAR2(1)	Status of the backup file, either U (unavailable) or X (expired)
FILESIZE	NUMBER	Size of the file
DEVICE_TYPE	VARCHAR2(17)	Type of device on which the file resides
FILENAME	VARCHAR2(513)	Name of the file
MEDIA	VARCHAR2(65)	Name of the media on which the copy resides. This value is informational only. It is not needed for restore.
MEDIA_POOL	NUMBER	Media pool in which the copy resides. This is the same value that was entered in the POOL operand of the Recovery Manager BACKUP command.

V$VERSION

V$VERSION displays version numbers of core library components in the Oracle Database. There is one row for each component.

Column	Datatype	Description
BANNER	VARCHAR2(80)	Component name and version number

V$VPD_POLICY

V$VPD_POLICY displays all the fine-grained security policies and predicates associated with the cursors currently in the library cache.

Column	Datatype	Description
ADDRESS	RAW(4 | 8)	Cursor address
PARADDR	RAW(4 | 8)	Parent cursor address
SQL_HASH	NUMBER	SQL hash number
SQL_ID	VARCHAR2(13)	SQL identifier
CHILD_NUMBER	NUMBER	Cursor's child number under the parent
OBJECT_OWNER	VARCHAR2(30)	Owner of the object with the policy
OBJECT_NAME	VARCHAR2(30)	Name of the object with the policy
POLICY_GROUP	VARCHAR2(30)	Name of the policy group
POLICY	VARCHAR2(30)	Name of the policy
POLICY_FUNCTION_OWNER	VARCHAR2(30)	Owner of the policy function
PREDICATE	VARCHAR2(4000)	Predicate for the policy (truncated to 4000 bytes in length)

V$WAIT_CHAINS

V$WAIT_CHAINS displays information about blocked sessions. A wait chain is comprised of sessions that are blocked by one another. Each row represents a blocked and blocker session pair. If a wait chain is not a cyclical wait chain, then the last row for the chain does not have a blocker.

Column	Datatype	Description
CHAIN_ID	NUMBER	A number identifying the wait chain
CHAIN_IS_CYCLE	VARCHAR2(5)	Indicates whether the final blocked session in the wait chain is blocked by the initial blocked session (TRUE) or not (FALSE)
CHAIN_SIGNATURE	VARCHAR2(801)	An Oracle-specific text signature of the wait chain. This signature can be used to identify similar wait chains.
CHAIN_SIGNATURE_HASH	NUMBER	A numeric representation of CHAIN_SIGNATURE
INSTANCE	NUMBER	Blocked session's instance identifier
OSID	VARCHAR2(25)	Blocked session's operating system process identifier
PID	NUMBER	Blocked session's Oracle process identifier
SID	NUMBER	Blocked session's Oracle session identifier
SESS_SERIAL#	NUMBER	Blocked session's Oracle session serial number
BLOCKER_IS_VALID	VARCHAR2(5)	Indicates whether the blocked session has a blocker (TRUE) or not (FALSE)
BLOCKER_INSTANCE	NUMBER	Blocker session's instance identifier; NULL if BLOCKER_IS_VALID is FALSE
BLOCKER_OSID	VARCHAR2(25)	Blocker session's operating system process identifier; NULL if BLOCKER_IS_VALID is FALSE
BLOCKER_PID	NUMBER	Blocker session's Oracle process identifier; NULL if BLOCKER_IS_VALID is FALSE
BLOCKER_SID	NUMBER	Blocker session's Oracle session identifier; NULL if BLOCKER_IS_VALID is FALSE
BLOCKER_SESS_SERIAL#	NUMBER	Blocker session's Oracle session serial number; NULL if BLOCKER_IS_VALID is FALSE
BLOCKER_CHAIN_ID	NUMBER	If not NULL, then the blocker session is a member of another chain specified by this chain identifier. For the remaining wait chain information, see the wait chain with the specified CHAIN_ID.
IN_WAIT	VARCHAR2(5)	Indicates whether the blocked session is in a wait (TRUE) or not (FALSE)
TIME_SINCE_LAST_WAIT_SECS	NUMBER	Number of seconds since the last time the blocked session waited; NULL if IN_WAIT is TRUE
WAIT_ID	NUMBER	A number identifying the wait; NULL if IN_WAIT is FALSE
WAIT_EVENT	NUMBER	Resource or event number for which the blocked session is waiting; NULL if IN_WAIT is FALSE
WAIT_EVENT_TEXT	VARCHAR2(64)	Resource or event for which the blocked session is waiting; NULL if IN_WAIT is FALSE
P1	NUMBER	First additional wait parameter; NULL if IN_WAIT is FALSE
P1_TEXT	VARCHAR2(64)	Description of the first additional wait parameter; NULL if IN_WAIT is FALSE
P2	NUMBER	Second additional wait parameter; NULL if IN_WAIT is FALSE
P2_TEXT	VARCHAR2(64)	Description of the second additional wait parameter; NULL if IN_WAIT is FALSE
P3	NUMBER	Third additional wait parameter; NULL if IN_WAIT is FALSE
P3_TEXT	VARCHAR2(64)	Description of the third additional wait parameter; NULL if IN_WAIT is FALSE
IN_WAIT_SECS	NUMBER	Seconds the blocked session has been in the current wait; NULL if IN_WAIT is FALSE
TIME_REMAINING_SECS	NUMBER	Seconds remaining until the blocked session ends its wait (-1 if the blocked session can indefinitely wait); NULL if IN_WAIT is FALSE
NUM_WAITERS	NUMBER	Number of sessions waiting for the blocked session
ROW_WAIT_OBJ#	NUMBER	Object ID for the table containing the row specified in ROW_WAIT_ROW#; NULL if IN_WAIT is FALSE

Column	Datatype	Description
ROW_WAIT_FILE#	NUMBER	Identifier for the datafile containing the row specified in ROW_WAIT_ROW#; NULL if IN_WAIT is FALSE. This column is valid only if the blocked session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_BLOCK#	NUMBER	Identifier for the block containing the row specified in ROW_WAIT_ROW#; NULL if IN_WAIT is FALSE. This column is valid only if the blocked session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.
ROW_WAIT_ROW#	NUMBER	Current row being locked; NULL if IN_WAIT is FALSE. This column is valid only if the blocked session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is not -1.

V$WAITCLASSMETRIC

V$WAITCLASSMETRIC displays metric values of wait classes for the most recent 60-second interval. A history of the last one hour will be kept in the system.

Column	Datatype	Description
BEGIN_TIME	DATE	Begin time of the interval
END_TIME	DATE	End time of the interval
INTSIZE_CSEC	NUMBER	Interval size (in hundredths of a second)
WAIT_CLASS#	NUMBER	Number of the class of the wait event
WAIT_CLASS_ID	NUMBER	Identifier of the class of the wait event
AVERAGE_WAITER_COUNT	NUMBER	Average waiter count
DBTIME_IN_WAIT	NUMBER	Percent of database time spent in the wait
TIME_WAITED	NUMBER	Time waited during the interval (in microseconds)
WAIT_COUNT	NUMBER	Number of times waited

V$WAITCLASSMETRIC_HISTORY

V$WAITCLASSMETRIC_HISTORY displays metric values of wait classes for all intervals in the last one hour.

The columns for V$WAITCLASSMETRIC_HISTORY are the same as those for V$WAITCLASSMETRIC.

See Also: "V$WAITCLASSMETRIC" on page 8-83

V$WAITSTAT

V$WAITSTAT displays block contention statistics. This table is only updated when timed statistics are enabled.

Column	Datatype	Description
CLASS	VARCHAR2(18)	Class of the block
COUNT	NUMBER	Number of waits by this OPERATION for this CLASS of block
TIME	NUMBER	Sum of all wait times for all the waits by this OPERATION for this CLASS of block

V$WALLET

V$WALLET displays metadata of certificates that may be used as a master key for transparent data encryption.

Column	Datatype	Description
CERT_ID	VARCHAR2(52)	A unique certificate identifier value used to specify a particular PKI certificate for use as the master key
DN	VARCHAR2(255)	Distinguished name of a particular PKI certificate
SERIAL_NUM	VARCHAR2(40)	Unique serial number assigned to a certificate by the issuer or signer
ISSUER	VARCHAR2(255)	Distinguished name of the Certificate Authority or issuer that issued and signed the certificate
KEYSIZE	NUMBER	Size of the PKI key associated with the certificate
STATUS	VARCHAR2(16)	Current status of the certificate: ■ UNUSED ■ IN USE ■ USED This column allows the user to identify whether a certificate is currently in use or has already been used for transparent database encryption.

V$WORKLOAD_REPLAY_THREAD

V$WORKLOAD_REPLAY_THREAD displays information for all the different types of replay sessions.

Column	Datatype	Description
SID	NUMBER	Session ID of the replay session
SERIAL#	NUMBER	Session serial number of the replay session
SPID	VARCHAR2(24)	The server process ID of the replay session
LOGON_USER	VARCHAR2(30)	Logon username of the replay session
LOGON_TIME	DATE	Logon time of the replay session
EVENT	VARCHAR2(64)	Event name
EVENT_ID	NUMBER	Event identifier
EVENT#	NUMBER	Event operation code
P1TEXT	VARCHAR2(64)	Text for event parameter 1
P1	NUMBER	Value of event parameter 1
P2TEXT	VARCHAR2(64)	Text for event parameter 2
P2	NUMBER	Value of event parameter 2
P3TEXT	VARCHAR2(64)	Text for event parameter 3
P3	NUMBER	Value of event parameter 3
WAIT_FOR_SCN	NUMBER	The captured SCN for which the current user call should wait
FILE_ID	NUMBER	The stream ID that is being replayed
CALL_COUNTER	NUMBER	The call counter of the user call that is being replayed
DEPENDENT_SCN	NUMBER	The dependent SCN, captured to order the commit actions using block-level dependencies
STATEMENT_SCN	NUMBER	Statement SCN
COMMIT_WAIT_SCN	NUMBER	The (maximum) SCN that the current commit should wait for
POST_COMMIT_SCN	NUMBER	The next SCN after the current commit

Column	Datatype	Description
ACTION_TYPE	NUMBER	The type of commits. The possible values are: COMMIT, ROLLBACK, FAKED_COMMIT, and NULL. A value of NULL means it is not a commit operation.
SESSION_TYPE	VARCHAR2(13)	The type of replay session: LOGON ADMIN DISPATCHER REPLAY
WRC_ID	NUMBER	Unique replay client ID assigned by the server to all participating replay clients when replay starts
FILE_NAME	VARCHAR2(30)	File name of the captured stream
SKIP_IT	VARCHAR2(1)	Whether or not the current replayed user call is skipped or not
DIRTY_BUFFERS	VARCHAR2(1)	???
DBTIME	NUMBER	Accumulated database time for the replay session
NETWORK_TIME	NUMBER	Accumulated network time for the replay session
THINK_TIME	NUMBER	Accumulated think time for the replay session
TIME_GAIN	NUMBER	If nonzero, the accumulated time in macro seconds indicating how fast the replay is
TIME_LOSS	NUMBER	If nonzero, the accumulated time in macro seconds indicating how slow the replay is
USER_CALLS	NUMBER	Total number of user calls
NEW_ERRORS	NUMBER	Total number of new errors
MUTATED_ERRORS	NUMBER	Total number of different errors
CLIENT_OS_USER	VARCHAR2(15)	Operating system username of the replay client
CLIENT_HOST	VARCHAR2(64)	Host name of the replay client
CLIENT_PID	VARCHAR2(24)	Process ID of the replay client
PROGRAM	VARCHAR2(48)	Program name of the replay client

V$XML_AUDIT_TRAIL

V$XML_AUDIT_TRAIL shows standard, fine-grained, SYS, and mandatory audit records written in XML format files.

Column	Datatype	Description
AUDIT_TYPE	NUMBER	Type of audit row. The possible values are: ▪ 1 = Standard XML Audit ▪ 2 = Fine Grained XML Audit ▪ 3 = SYS XML Audit ▪ 4 = Mandatory XML Audit
SESSION_ID	NUMBER	Numeric ID for the Oracle session
PROXY_SESSIONID	NUMBER	Proxy session serial number, if an enterprise user has logged in through a proxy mechanism
STATEMENTID	NUMBER	Numeric ID for the statement run (a statement may cause multiple audit records)
ENTRYID	NUMBER	Numeric ID for the audit trail entry in the session
EXTENDED_TIMESTAMP	TIMESTAMP(6) WITH TIME ZONE	Timestamp of the audited operation (the timestamp of the user's logon for entries is created by AUDIT SESSION)
GLOBAL_UID	VARCHAR2(32)	Global user identifier for the user, if the user has logged in as an enterprise user

Column	Datatype	Description
DB_USER	VARCHAR2(30)	Database username of the user whose actions were audited
CLIENTIDENTIFIER	VARCHAR2(64)	Client identifier in the Oracle session
EXT_NAME	VARCHAR2(1024)	User's external name
OS_USER	VARCHAR2(30)	Operating system logon user name of the user whose actions were audited
OS_HOST	VARCHAR2(128)	Client host machine name
OS_PROCESS	VARCHAR2(16)	Operating system process identifier of the Oracle server process
TERMINAL	VARCHAR2(30)	Identifier for the user's terminal
INSTANCE_NUMBER	NUMBER	Instance number as specified in the initialization parameter file, init.ora
OBJECT_SCHEMA	VARCHAR2(30)	Owner of the audited object
OBJECT_NAME	VARCHAR2(30)	Name of the object affected by the action
POLICY_NAME	VARCHAR2(30)	Name of fine-grained auditing policy
NEW_OWNER	VARCHAR2(30)	Owner of the object named in the NEW_NAME column
NEW_NAME	VARCHAR2(30)	New name of object after renaming, or the name of an underlying object (for example, CREATE INDEX owner.obj_name ON new_owner.new_name)
ACTION	NUMBER	Numeric code for the action type
STATEMENT_TYPE	NUMBER	Description of the action
TRANSACTIONID	RAW(8)	Identifier of the transaction in which the object is accessed or modified
RETURNCODE	NUMBER	Oracle error code generated by the action. Zero if the action succeeded.
SCN	NUMBER	System change number (SCN) of the query
COMMENT_TEXT	VARCHAR2(4000)	Text comments on standard audit entries. Also indicates how the user was authenticated - the method can be one of the following: ■ DATABASE - authentication was done by password ■ NETWORK - authentication was done by Net8 or the Advanced Networking Option ■ PROXY - the client was authenticated by another user. The name of the proxy user follows the method type.
AUTH_PRIVILEGES	VARCHAR2(16)	Privileges granted and revoked in GRANT and REVOKE statements recorded for standard audit trail entry
GRANTEE	VARCHAR2(30)	User who granted or revoked the privilege
PRIV_USED	NUMBER	Numerical code of privileges, if any, used in the action
SES_ACTIONS	VARCHAR2(16)	Session summary for standard audit records. A string of 12 characters, one for each action type, in the following order: Alter, Audit, Comment, Delete, Grant, Index, Insert, Lock, Rename, Select, Update, Flashback. Values: - = None, S=Success, F=Failure, B=Both
OS_PRIVILEGE	VARCHAR2(7)	Operating privilege (SYSDBA or SYSOPER), if any, used in the session. If no privilege is used, it will be NONE.
ECONTEXT_ID	VARCHAR2(64)	Application execution context identifier
SQL_BIND	VARCHAR2(4000)	List of bind variables used in the statement
SQL_TEXT	VARCHAR2(4000)	The statement or command that triggered the audit event
OBJECT_EDITION	VARCHAR2(30)	???

V$XS_SESSION

V$XS_SESSION displays information about the state of the XS$SESSION and XS$GLOBAL_VAR namespaces for all lightweight user sessions in the database.

Column	Datatype	NULL	Description
USERNAME	VARCHAR2(4000)	NOT NULL	Name of the lightweight user associated with the lightweight user session
USERID	RAW(16)	NOT NULL	Global unique identifier of the lightweight user associated with the lightweight user session
ACLOID	RAW(16)		UUID of the ACL on the lightweight user session
COOKIE	VARCHAR2(1024)		Server-unique cookie value associated with the session
PROXYID	RAW(16)		UUID of the previous XS user who, through the SWITCH_USER API, has proxied into the current lightweight user identity
CREATORID	RAW(16)	NOT NULL	UUID of the user that created the lightweight user session
UPDATEID	RAW(16)	NOT NULL	UUID of the user that last accessed the lightweight user session
CREATETIME	TIMESTAMP(6)	NOT NULL	Creation time for the lightweight user session
LAST_AUTH	TIMESTAMP(6)	NOT NULL	Last time the lightweight user was authenticated **Note:** The application or middle-tier must call the REAUTH_SESSION PL/SQL procedure or the reauthenticateSession Java method to change this value.
LAST_ACCESS	TIMESTAMP(6)	NOT NULL	Time when the lightweight user session was last accessed
TIMEOUT	NUMBER(6)		Amount of time (in minutes) before the session is considered to have timed out
NLS_CALENDAR	VARCHAR2(255)		NLS_CALENDAR parameter corresponding to the lightweight user session
NLS_COMP	VARCHAR2(255)		NLS_COMP parameter corresponding to the lightweight user session
NLS_CURRENCY	VARCHAR2(255)		NLS_CURRENCY parameter corresponding to the lightweight user session
NLS_DATE_FORMAT	VARCHAR2(255)		NLS_DATE_FORMAT parameter corresponding to the lightweight user session
NLS_DATE_LANGUAGE	VARCHAR2(255)		NLS_DATE_LANGUAGE parameter corresponding to the lightweight user session
NLS_ISO_CURRENCY	VARCHAR2(255)		NLS_ISO_CURRENCY parameter corresponding to the lightweight user session
NLS_LANGUAGE	VARCHAR2(255)		NLS_LANGUAGE parameter corresponding to the lightweight user session
NLS_LENGTH_SEMANTICS	VARCHAR2(255)		NLS_LENGTH_SEMANTICS parameter corresponding to the lightweight user session
NLS_NCHAR_CONV_EXCP	VARCHAR2(255)		NLS_NCHAR_CONV_EXCP parameter corresponding to the lightweight user session
NLS_NUMERIC_CHRS	VARCHAR2(255)		NLS_NUMERIC_CHARACTERS parameter corresponding to the lightweight user session
NLS_SORT	VARCHAR2(255)		NLS_SORT parameter corresponding to the lightweight user session
NLS_TERRITORY	VARCHAR2(255)		NLS_TERRITORY parameter corresponding to the lightweight user session
NLS_TIMESTAMP_FMT	VARCHAR2(255)		NLS_TIMESTAMP_FORMAT parameter corresponding to the lightweight user session
NLS_TIMESTAMP_TZ_FMT	VARCHAR2(255)		NLS_TIMESTAMP_TZ_FORMAT parameter corresponding to the lightweight user session
NLS_DUAL_CURRENCY	VARCHAR2(255)		NLS_DUAL_CURRENCY parameter corresponding to the lightweight user session

Column	Datatype	NULL	Description
APPS_FEATURE	VARCHAR2(255)		???

V$XS_SESSION_ATTRIBUTE

V$XS_SESSION_ATTRIBUTE displays information about the namespaces and attributes in all lightweight user sessions in the database as of the end of the last request. The state of any active request is not reflected in this view.

Column	Datatype	NULL	Description
LWSID	RAW(16)	NOT NULL	Lightweight session ID
NAMESPACE	VARCHAR2(4000)	NOT NULL	Namespace name
NAME	VARCHAR2(4000)		Attribute name
VALUE	VARCHAR2(4000)		Attribute value
ACLOID	RAW(16)		ACL OID for the namespace
EVENT_HANDLER	VARCHAR2(255)		Namespace handler

V$XS_SESSION_ROLE

V$XS_SESSION_ROLE displays information about the enabled roles for all lightweight user sessions in the database as of the end of the last request. The security context of active requests is not reflected in this view.

Column	Datatype	NULL	Description
NAME	VARCHAR2(4000)	NOT NULL	Role name
ROLEID	NUMBER(10)	NOT NULL	Role Internal ID
UUID	RAW(16)	NOT NULL	Role ID
LWSID	RAW(16)	NOT NULL	Lightweight session ID

A

Database Limits

This appendix lists the limits of values associated with database functions and objects. Limits exist on several levels in the database. There is usually a hard-coded limit in the database that cannot be exceeded. This value may be further restricted for any given operating system.

Database limits are divided into four categories:

- Datatype Limits
- Physical Database Limits
- Logical Database Limits
- Process and Runtime Limits

See Also: Your operating system-specific Oracle documentation

Datatype Limits

Datatypes	Limit	Comments
BFILE	Maximum size: 4 GB Maximum size of a file name: 255 characters Maximum size of a directory name: 30 characters Maximum number of open BFILEs: see Comments	The maximum number of BFILEs is limited by the value of the SESSION_MAX_OPEN_FILES initialization parameter, which is itself limited by the maximum number of open files the operating system will allow.
BLOB	Maximum size: (4 GB - 1) * DB_BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000[1]).
CHAR	Maximum size: 2000 bytes	None
CHAR VARYING	Maximum size: 4000 bytes	None
CLOB	Maximum size: (4 GB - 1) * DB_BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000[1]).
Literals (characters or numbers in SQL or PL/SQL)	Maximum size: 4000 characters	None
LONG	Maximum size: 2 GB - 1	Only one LONG column is allowed per table.
NCHAR	Maximum size: 2000 bytes	None

Datatypes	Limit	Comments
NCHAR VARYING	Maximum size: 4000 bytes	None
NCLOB	Maximum size: (4 GB - 1) * DB_BLOCK_SIZE initialization parameter (8 TB to 128 TB)	The number of LOB columns per table is limited only by the maximum number of columns per table (that is, 1000[1]).
NUMBER	999...(38 9's) x10^{125} maximum value	Can be represented to full 38-digit precision (the mantissa)
	-999...(38 9's) x10^{125} minimum value	Can be represented to full 38-digit precision (the mantissa)
Precision	38 significant digits	None
RAW	Maximum size: 2000 bytes	None
VARCHAR	Maximum size: 4000 bytes	None
VARCHAR2	Maximum size: 4000 bytes	None

[1] The absolute maximum number of columns in a table is 1000. However, when you create an object table (or a relational table with columns of object, nested table, varray, or REF type), Oracle maps the columns of the user-defined types to relational columns, creating in effect hidden columns that count toward the 1000-column limit. For details on how Oracle calculates the total number of columns in such a table, refer to *Oracle Database Administrator's Guide*.

See Also: *Oracle Database SQL Language Reference* for information on the rules and limitations governing the use of datatypes and naming database objects

Physical Database Limits

Item	Type of Limit	Limit Value
Database Block Size	Minimum	2048 bytes; must be a multiple of operating system physical block size
Database Block Size	Maximum	Operating system dependent; never more than 32 KB
Database Blocks	Minimum in initial extent of a segment	2 blocks
Database Blocks	Maximum per datafile	Platform dependent; typically 2^{22} - 1 blocks
Controlfiles	Number of control files	1 minimum; 2 or more (on separate devices) strongly recommended
Controlfiles	Size of a control file	Dependent on operating system and database creation options; maximum of 20,000 x (database block size)
Database files	Maximum per tablespace	Operating system dependent; usually 1022
Database files	Maximum per database	65533 May be less on some operating systems Limited also by size of database blocks and by the DB_FILES initialization parameter for a particular instance
Database extents	Maximum per dictionary managed tablespace	4 GB * physical block size (with K/M modifier); 4 GB (without K/M modifier)
Database extents	Maximum per locally managed (uniform) tablespace	2 GB * physical block size (with K/M modifier); 2 GB (without K/M modifier)
Database file size	Maximum	Operating system dependent. Limited by maximum operating system file size; typically 2^{22} or 4 MB blocks

Item	Type of Limit	Limit Value
MAXEXTENTS	Default value	Derived from tablespace default storage or DB_BLOCK_SIZE initialization parameter
MAXEXTENTS	Maximum	Unlimited
Redo Log Files	Maximum number of logfiles	Limited by value of MAXLOGFILES parameter in the CREATE DATABASE statement Control file can be resized to allow more entries; ultimately an operating system limit
Redo Log Files	Maximum number of logfiles per group	Unlimited
Redo Log File Size	Minimum size	4 MB
Redo Log File Size	Maximum Size	Operating system limit; typically 2 GB
Tablespaces	Maximum number per database	64 K Number of tablespaces cannot exceed the number of database files because each tablespace must include at least one file
Bigfile Tablespaces	Number of blocks	A bigfile tablespace contains only one datafile or tempfile, which can contain up to approximately 4 billion (2^{32}) blocks. The maximum size of the single datafile or tempfile is 128 terabytes (TB) for a tablespace with 32 K blocks and 32 TB for a tablespace with 8 K blocks.
Smallfile (traditional) Tablespaces	Number of blocks	A smallfile tablespace is a traditional Oracle tablespace, which can contain 1022 datafiles or tempfiles, each of which can contain up to approximately 4 million (2^{22}) blocks.
External Tables file	Maximum size	Dependent on the operating system. An external table can be composed of multiple files.

Logical Database Limits

Item	Type of Limit	Limit Value
GROUP BY clause	Maximum length	The GROUP BY expression and all of the nondistinct aggregate functions (for example, SUM, AVG) must fit within a single database block.
Indexes	Maximum per table	Unlimited
Indexes	Total size of indexed column	75% of the database block size minus some overhead
Columns	Per table	1000 columns maximum
Columns	Per index (or clustered index)	32 columns maximum
Columns	Per bitmapped index	30 columns maximum
Constraints	Maximum per column	Unlimited
Subqueries	Maximum levels of subqueries in a SQL statement	Unlimited in the FROM clause of the top-level query 255 subqueries in the WHERE clause
Partitions	Maximum length of linear partitioning key	4 KB - overhead

Item	Type of Limit	Limit Value
Partitions	Maximum number of columns in partition key	16 columns
Partitions	Maximum number of partitions allowed per table or index	1024K - 1
Rows	Maximum number per table	Unlimited
Stored Packages	Maximum size	PL/SQL and Developer/2000 may have limits on the size of stored procedures they can call. The limits typically range from 2000 to 3000 lines of code. **See Also:** Your PL/SQL or Developer/2000 documentation for details
Trigger Cascade Limit	Maximum value	Operating system-dependent, typically 32
Users and Roles	Maximum	2,147,483,638
Tables	Maximum per clustered table	32 tables
Tables	Maximum per database	Unlimited

Note: The limit on how long a SQL statement can be depends on many factors, including database configuration, disk space, and memory

Note: When an object instance exists in memory, there is no fixed limit on the number of attributes in the object. But the maximum total amount of memory consumed by an object instance is 4 GB.

When an object instance is inserted into a table, the attributes are exploded into separate columns in the table, and the Oracle 1000-column limit applies.

Process and Runtime Limits

Item	Type of Limit	Limit Value
Instances per database	Maximum number of cluster database instances per database	Operating system-dependent
Locks	Row-level	Unlimited
Locks	Distributed Lock Manager	Operating system dependent
SGA size	Maximum value	Operating system-dependent; typically 2 to 4 GB for 32-bit operating systems, and > 4 GB for 64-bit operating systems
Advanced Queuing Processes	Maximum per instance	10
Job Queue Processes	Maximum per instance	1000
I/O Slave Processes	Maximum per background process (DBWR, LGWR, etc.)	15
I/O Slave Processes	Maximum per Backup session	15
Sessions	Maximum per instance	32 KB; limited by the PROCESSES and SESSIONS initialization parameters
Global Cache Service Processes	Maximum per instance	10

Item	Type of Limit	Limit Value
Shared Servers	Maximum per instance	Unlimited within constraints set by the PROCESSES and SESSIONS initialization parameters, for instance
Dispatchers	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance
Parallel Execution Slaves	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance
Backup Sessions	Maximum per instance	Unlimited within constraints set by PROCESSES and SESSIONS initialization parameters, for instance

B

SQL Scripts

This appendix describes SQL scripts that are required for optimal operation of the Oracle Database.

The SQL scripts are described in the following sections:

- Creating the Data Dictionary
- Creating Additional Data Dictionary Structures
- The "NO" Scripts
- Upgrade and Downgrade Scripts
- Java Scripts

> **Note:** Comments within the SQL scripts themselves contain more detailed information and examples.

Creating the Data Dictionary

When you use the Database Configuration Assistant to create a database, Oracle automatically creates the data dictionary. Thereafter, whenever the database is in operation, Oracle updates the data dictionary in response to every DDL statement.

The data dictionary base tables are the first objects created in any Oracle database. They are created in the `SYSTEM` tablespace and must remain there. The data dictionary base tables store information about all user-defined objects in the database.

Table B–1 lists required scripts, which are run automatically when you create a database using the Database Configuration Assistant. They are described here because you might need to run them if you create a database manually. To run these scripts, you must be connected to Oracle as a user with `SYSDBA` privileges.

Table B–1 Creating the Data Dictionary Scripts

Script Name	Needed For	Description
`catalog.sql`	All databases	Creates the data dictionary and public synonyms for many of its views Grants `PUBLIC` access to the synonyms
`catproc.sql`	All databases	Runs all scripts required for, or used with, PL/SQL
`catclust.sql`	Real Application Clusters	Creates Real Application Clusters data dictionary views

See Also:

- Your operating system-specific Oracle documentation for the exact names and locations of these scripts on your operating system
- *Oracle Database Administrator's Guide* for more information about creating a database
- Your release notes and *Oracle Database Upgrade Guide* to learn when it is necessary to run these scripts

Creating Additional Data Dictionary Structures

Oracle supplies other scripts that create additional structures you can use in managing your database and creating database applications. These scripts are listed in Table B–2.

See Also: Your operating system-specific Oracle documentation for the exact names and locations of these scripts on your operating system

Table B–2 Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
`catblock.sql`	Performance management	`SYS`	Creates views that can dynamically display lock dependency graphs
`catexp7.sql`	Exporting data to Oracle7	`SYS`	Creates the dictionary views needed for the Oracle7 Export utility to export data from the Oracle Database in Oracle7 Export file format
`caths.sql`	Heterogeneous Services	`SYS`	Installs packages for administering heterogeneous services
`catio.sql`	Performance management	`SYS`	Allows I/O to be traced on a table-by-table basis
`catoctk.sql`	Security	`SYS`	Creates the Oracle Cryptographic Toolkit package
`catqueue.sql`	Advanced Queuing		Creates the dictionary objects required for Advanced Queuing
`catrep.sql`	Oracle Replication	`SYS`	Runs all SQL scripts for enabling database replication
`dbmsiotc.sql`	Storage management	Any user	Analyzes chained rows in index-organized tables
`dbmspool.sql`	Performance management	`SYS` or `SYSDBA`	Enables DBA to lock PL/SQL packages, SQL statements, and triggers into the shared pool
`userlock.sql`	Concurrency control	`SYS` or `SYSDBA`	Provides a facility for user-named locks that can be used in a local or clustered environment to aid in sequencing application actions
`utlbstat.sql` and `utlestat.sql`	Performance monitoring	`SYS`	Respectively start and stop collecting performance tuning statistics
`utlchn1.sql`	Storage management	Any user	For use with the Oracle Database. Creates tables for storing the output of the `ANALYZE` command with the `CHAINED ROWS` option. Can handle both physical and logical rowids.

Table B–2 (Cont.) Creating Additional Data Dictionary Structures

Script Name	Needed For	Run By	Description
utlconst.sql	Year 2000 compliance	Any user	Provides functions to validate that CHECK constraints on date columns are year 2000 compliant
utldtree.sql	Metadata management	Any user	Creates tables and views that show dependencies between objects
utlexpt1.sql	Constraints	Any user	For use with the Oracle Database. Creates the default table (EXCEPTIONS) for storing exceptions from enabling constraints. Can handle both physical and logical rowids.
utlip.sql	PL/SQL	SYS	Used primarily for upgrade and downgrade operations. It invalidates all existing PL/SQL modules by altering certain dictionary tables so that subsequent recompilations will occur in the format required by the database. It also reloads the packages STANDARD and DBMS_STANDARD, which are necessary for any PL/SQL compilations.
utlirp.sql	PL/SQL	SYS	Used to change from 32-bit to 64-bit word size or vice versa. This script recompiles existing PL/SQL modules in the format required by the new database. It first alters some data dictionary tables. Then it reloads the packages STANDARD and DBMS_STANDARD, which are necessary for using PL/SQL. Finally, it triggers a recompilation of all PL/SQL modules, such as packages, procedures, and types.
utllockt.sql	Performance monitoring	SYS or SYSDBA	Displays a lock wait-for graph, in tree structure format
utlpwdmg.sql	Security	SYS or SYSDBA	Creates PL/SQL functions for default password complexity verification. Sets the default password profile parameters and enables password management features.
utlrp.sql	PL/SQL	SYS	Recompiles all existing PL/SQL modules that were previously in an INVALID state, such as packages, procedures, and types.
utlsampl.sql	Examples	SYS or any user with DBA role	Creates sample tables, such as emp and dept, and users, such as scott
utlscln.sql	Oracle Replication	Any user	Copies a snapshot schema from another snapshot site
utltkprf.sql	Performance management	SYS	Creates the TKPROFER role to allow the TKPROF profiling utility to be run by non-DBA users
utlvalid.sql	Partitioned tables	Any user	Creates tables required for storing output of ANALYZE TABLE ...VALIDATE STRUCTURE of a partitioned table
utlxplan.sql	Performance management	Any user	Creates the table PLAN_TABLE, which holds output from the EXPLAIN PLAN statement

The "NO" Scripts

The scripts in Table B–3 are used to remove dictionary information for various optional services or components.

> **See Also:** *Oracle Database Upgrade Guide* for more information about these scripts

Table B–3 The NO Scripts

Script Name	Needed For	Run By	Description
`catnoadt.sql`	Objects	`SYS`	Drops views and synonyms on dictionary metadata that relate to object types
`catnoaud.sql`	Security	`SYS`	Drops views and synonyms on auditing metadata
`catnohs.sql`	Heterogeneous Services	`SYS`	Removes Heterogeneous Services dictionary metadata
`catnoprt.sql`	Partitioning	`SYS`	Drops views and synonyms on dictionary metadata that relate to partitioned tables and indexes
`catnosvm.sql`	Server Manager	`SYS`	Removes Oracle7 Server Manager views and synonyms
`catnsnmp.sql`	Distributed management	`SYS`	Drops the DBSNMP user and SNMPAGENT role

Upgrade and Downgrade Scripts

The scripts in Table B–4 are used when upgrading or downgrading to another release of Oracle. To run these scripts, you must be connected to Oracle as a user with `SYSDBA` privileges.

Table B–4 Upgrade and Downgrade Scripts

Script Name	Needed For	Description
`catdwgrd.sql`	Downgrading	Provides a direct downgrade path from the new Oracle Database 10*g* release
`catupgrd.sql`	Upgrading	Provides a direct upgrade path to the new Oracle Database 10*g* release
`utlu111i.SQL`	Pre-Upgrade Information	Analyzes the database to be upgraded, detailing requirements and issues for the upgrade to release 11.1
`utlu111s.SQL`	Post-Upgrade Status	Displays the component upgrade status after an upgrade to release 11.1

> **See Also:** *Oracle Database Upgrade Guide* for more information about these scripts

Java Scripts

The scripts in Table B–5 are useful only if the JServer option is installed.

Table B–5 Java Scripts

Script Name	Description
`initjvm.sql`	Initializes JServer by installing core Java class libraries and Oracle-specific Java classes
`rmjvm.sql`	Removes all elements of the JServer
`catjava.sql`	Installs Java-related packages and classes

See Also: *Oracle Database Java Developer's Guide*

C

Oracle Wait Events

This appendix contains the following topics:

- Classes of Wait Events
- Descriptions of Common Wait Event Parameters
- Descriptions of Wait Events

Information about wait events is displayed in three dynamic performance views:

- V$SESSION_WAIT displays the events for which sessions have just completed waiting or are currently waiting.
- V$SYSTEM_EVENT displays the total number of times all the sessions have waited for the events in that view.
- V$SESSION_EVENT is similar to V$SYSTEM_EVENT, but displays all waits for each session.

> **See Also:** "V$SESSION_EVENT" on page 8-15, "V$SESSION_WAIT" on page 8-18, and "V$SYSTEM_EVENT" on page 8-68

Many of these wait events are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice. Application developers should be aware of this and write their code to tolerate missing or extra wait events.

The following SQL statement displays an alphabetical list of all Oracle wait events and the wait class to which they belong:

```
SQL> SELECT name, wait_class FROM V$EVENT_NAME ORDER BY name;
```

Classes of Wait Events

Every wait event belongs to a class of wait event. The following list describes each of the wait classes.

Administrative

Waits resulting from DBA commands that cause users to wait (for example, an index rebuild)

Application

Waits resulting from user application code (for example, lock waits caused by row level locking or explicit lock commands)

Cluster

Waits related to Real Application Cluster resources (for example, global cache resources such as 'gc cr block busy'

Commit

This wait class only comprises one wait event - wait for redo log write confirmation after a commit (that is, 'log file sync')

Concurrency

Waits for internal database resources (for example, latches)

Configuration

Waits caused by inadequate configuration of database or instance resources (for example, undersized log file sizes, shared pool size)

Idle

Waits that signify the session is inactive, waiting for work (for example, 'SQL*Net message from client')

Network

Waits related to network messaging (for example, 'SQL*Net more data to dblink')

Other

Waits which should not typically occur on a system (for example, 'wait for EMON to spawn')

Scheduler

Resource Manager related waits (for example, 'resmgr: become active')

System I/O

Waits for background process IO (for example, DBWR wait for 'db file parallel write')

User I/O

Waits for user IO (for example 'db file sequential read')

Descriptions of Common Wait Event Parameters

This section provides descriptions of some of the more common wait event parameters.

block#

This is the block number of the block for which Oracle needs to wait. The block number is relative to the start of the file. To find the object to which this block belongs, enter the following SQL statements:

```
select name, kind
from ext_to_obj_view
where file# = file#
     and lowb <= block#
     and highb >= block#;
```

blocks

The number of blocks that is being either read from or written to the file. The block size is dependent on the file type:

- Database files have a block size of DB_BLOCK_SIZE
- Logfiles and control files have a block size that is equivalent to the physical block size of the platform

break?

If the value for this parameter equals 0, a reset was sent to the client. A nonzero value indicates that a break was sent to the client.

class

The class of the block describes how the contents of the block are used. For example, class 1 represents data block, and class 4 represents segment header.

dba

The initials "dba" represents the data block address, which consists of a file number and a block number.

driver id

The address of the disconnect function of the driver that is currently being used.

file#

The following query returns the name of the database file:

```
select *
from v$datafile
where file# = file#;
```

id1

The first identifier (*id1*) of the enqueue or global lock takes its value from P2 or P2RAW. The meaning of the identifier depends on the name (P1).

id2

The second identifier (*id2*) of the enqueue or global lock takes its value from P3 or P3RAW. The meaning of the identifier depends on the name (P1).

le

The relative index number into V$GC_ELEMENT.

mode

The *mode* is usually stored in the low order bytes of P1 or P1RAW and indicates the mode of the enqueue or global lock request. This parameter has one of the following values:

Table C–1 Lock Mode Values

Mode Value	Description
1	Null mode
2	Sub-Share
3	Sub-Exclusive
4	Share
5	Share/Sub-Exclusive
6	Exclusive

Use the following SQL statement to retrieve the name of the lock and the mode of the lock request:

```
select chr(bitand(p1,-16777216)/16777215)||
       chr(bitand(p1, 16711680)/65535) "Lock",
       bitand(p1, 65535) "Mode"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

name and type

The name or "type" of the enqueue or global lock can be determined by looking at the two high order bytes of P1 or P1RAW. The name is always two characters. Use the following SQL statement to retrieve the lock name.

```
select chr(bitand(p1,-16777216)/16777215)||
      chr(bitand(p1,16711680)/65535) "Lock"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

namespace

The name of the object namespace as it is displayed in `V$DB_OBJECT_CACHE` view.

requests

The number of I/Os that are "requested." This differs from the number of blocks in that one request could potentially contain multiple blocks.

session#

The number of the inactive session. Use the following SQL statement to find more information about the session:

```
select *
from v$session
where sid = session#;
```

waited

This is the total amount of time the session has waited for this session to terminate.

Descriptions of Wait Events

This section provides detailed descriptions for those wait events of greatest interest. Where appropriate, pointers are provided to further information elsewhere in Oracle Database documentation. For a complete listing of wait events, in alphabetical order, you can issue the following SQL statement:

```
SQL> SELECT name FROM V$EVENT_NAME ORDER BY name;
```

alter system set dispatcher

A session has issued a statement `ALTER SYSTEM SET DISPATCHER` = `string` and is waiting for the dispatchers to get started.

Wait Time: The session will wait 1 / 100 of a second and check to see if the new dispatchers have started else the session will wait again

Parameter	Description
waited	Number of times that the session has waited 1 / 100 of a second

batched allocate scn lock request

A session is waiting on another process to allocate a system change number (SCN). If the foreground timed out waiting on a process to get the SCN, the foreground will get the SCN.

Wait Time: The wait time is 1 second on the assumption that an SCN allocation should normally need much less than that

Parameters: None

BFILE check if exists

The session waits to check if an external large object (LOB) exists.

Wait Time: The total elapsed time for the **exists** call

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE check if open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the **isopen** call

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE closure

The session waits for an external large object (LOB) to close.

Wait Time: The total elapsed time for the **close** call

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE get length

The session waits on a call to check the size of an external large object (LOB).

Wait Time: The total elapsed time for the call to check the LOB size

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE get name object

The session waits on a call to find or generate the external name of a external large object.

Wait Time: The total elapse time for **make external file name** to complete

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE get path object

The session is waiting on a call to find or generate the external path name of an external large object (LOB).

Wait Time: The total elapsed time for **make external path** to complete

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE internal seek

The session waits for a positioning call within the external large object (LOB) to complete.

Wait Time: The total elapse time for the **seek** to complete

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the **isopen** call

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

BFILE read

The session waits for a read from a external large object (LOB) to complete.

Wait Time: The total elapse time for the **read** to complete

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

broadcast mesg queue transition

Processes enter "wait for broadcast mesg queue transition" when cleaning up a publisher channel handle to a RELIABLE broadcast channel. The publisher is responsible for moving the message to the free queue, but it cannot do so until the message is in the done queue. If the message is still not in the done queue, process enters this wait. This wait event will most likely show up when an Oracle process is about to exit normally, or when PMON cleans up a dead process.

Wait Time: Varies

Parameter	Description
channel handle	publisher channel handle pointer
message	broadcast message pointer
location	A number indicating the function in KSR where the process is waiting

broadcast mesg recovery queue transition

Processes enter "wait for broadcast mesg recovery queue transition" when cleaning up a publisher channel handle to a RELIABLE broadcast channel. The broadcasted message is in the recovery queue of another channel handle (for example, ch2). Process enters this wait, if the message is yet to be removed from the recovery queue of the ch2 channel handle. This wait event will most likely show up when an Oracle process is about to exit normally, or when PMON cleans up a dead process.

Wait Time: Varies

Parameter	Description
channel handle	Publisher channel handle pointer
message	Broadcast message pointer
location	A number indicating the function in KSR where the process is waiting

buffer busy waits

Wait until a buffer becomes available. This event happens because a buffer is either being read into the buffer cache by another session (and the session is waiting for that read to complete) or the buffer is the buffer cache, but in a incompatible mode (that is, some other session is changing the buffer).

Wait Time: Normal wait time is 1 second. If the session was waiting for a buffer during the last wait, then the next wait will be 3 seconds.

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
id	The buffer busy wait event is called from different places in the session

buffer deadlock

Oracle does not really wait on this event; the foreground only yields the CPU. Thus, the chances of catching this event are very low. This is not an application induced deadlock, but an assumed deadlock by the cache layer. The cache layer cannot get a buffer in a certain mode within a certain amount of time.

Wait Time: 0 seconds. The foreground process only yields the CPU and will usually be placed at the end of the CPU run queue.

Parameter	Description
class	See "class" on page C-3
mode	See "mode" on page C-3
flag	The flag points to the internal flags used by the session to get this block
dba	See "dba" on page C-3

buffer latch

The session waits on the buffer hash chain latch. Primarily used in the dump routines.

Wait Time: 1 second

Parameter	Description
latch addr	The virtual address in the SGA where this latch is located. Use the following statement to find the name of this latch: `select * from v$latch a, v$latchname b where addr = latch addr and a.latch# = b.latch#;`
chain#	The index into array of buffer hash chains. When the chain is 0xfffffff, the foreground waits on the LRU latch.

buffer read retry

This event occurs only if the instance is mounted in shared mode (Oracle Real Application Cluster). During the read of the buffer, the contents changed. This means that either:

- The version number, dba, or the incarnation and sequence number stored in the block no longer match
- The checksum on the block does not match the checksum in the block

The block will be re-read (this may fail up to 3 times), then corruption is assumed and the corrupt block is dumped in the trace file.

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2

checkpoint completed

A session waits for a checkpoint to complete. This could happen, for example, during a close database or a local checkpoint.

Wait Time: 5 seconds

Parameters: None

cleanup of aborted processes

When a process spawn is aborted while the process spawning background is in the middle of spawning, the current session must wait until the pid of the new process is filled in. Once the pid is filled in, then the process spawn can be actually aborted.

Wait Time: Usually 3 seconds

Parameter	Description
location	Location of the wait

control file parallel write

This event occurs while the session is writing physical blocks to all control files. This happens when:

- The session starts a control file transaction (to make sure that the control files are up to date in case the session crashes before committing the control file transaction)
- The session commits a transaction to a control file
- Changing a generic entry in the control file, the new value is being written to all control files

Wait Time: The wait time is the time it takes to finish all writes to all control files

Parameter	Description
files	The number of control files to which the session is writing
blocks	The number of blocks that the session is writing to the control file
requests	The number of I/O requests which the session wants to write

control file sequential read

Reading from the control file. This happens in many cases. For example, while:

- Making a backup of the control files
- Sharing information (between instances) from the control file

- Reading other blocks from the control files
- Reading the header block

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
file#	The control file from which the session is reading
block#	Block number in the control file from where the session starts to read. The block size is the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read

control file single write

This wait is signaled while the control file's shared information is written to disk. This is an atomic operation protected by an enqueue (CF), so that only one session at a time can write to the entire database.

Wait Time: The wait time is the elapsed time of the write

Parameter	Description
file#	This identifies the control file to which the session is currently writing
block#	Block number in the control file where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read

cursor: mutex S

A session waits on this event when it is requesting a mutex in shared mode, when another session is currently holding a this mutex in exclusive mode on the same cursor object.

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contain SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

cursor: mutex X

The session requests the mutex for a cursor object in exclusive mode, and it must wait because the resource is busy. The mutex is busy because either the mutex is being held in exclusive mode by another session or the mutex is being held shared by one or more sessions. The existing mutex holder(s) must release the mutex before the mutex can be granted exclusively.

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contain SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)

Parameter	Description
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

cursor: pin S

A session waits on this event when it wants to update a shared mutex pin and another session is currently in the process of updating a shared mutex pin for the same cursor object. This wait event should rarely be seen because a shared mutex pin update is very fast.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

cursor: pin S wait on X

A session waits for this event when it is requesting a shared mutex pin and another session is holding an exclusive mutex pin on the same cursor object.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

cursor: pin X

A session waits on this event when it is requesting an exclusive mutex pin for a cursor object and it must wait because the resource is busy. The mutex pin for a cursor object can be busy either because a session is already holding it exclusive, or there are one or more sessions which are holding shared mutex pin(s). The exclusive waiter must wait until all holders of the pin for that cursor object have released it, before it can be granted.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

Data Guard: process clean up

During Data Guard process termination, Data Guard will wait for one second for process cleanup to complete.

Data Guard: process exit

During Data Guard process termination, Data Guard will wait for a process to exit before attempting any process cleanup that may be required. It will sleep for one second between each check for process exit.

Data Guard Broker: single instance

The Data Guard Broker (DMON) process waits for the other instances in this cluster to complete shutdown before continuing with the broker operation.

Wait Time: Depends on the number of instances, but not exceeding 30 seconds times the number of instances

db file parallel read

This happens during recovery. It can also happen during buffer prefetching, as an optimization (rather than performing multiple single-block reads). Database blocks that need to be changed as part of recovery are read in parallel from the database.

Wait Time: Wait until all of the I/Os are completed

Parameter	Description
files	This indicates the number of files to which the session is reading
blocks	This indicates the total number of blocks to be read
requests	This indicates the total number of I/O requests, which will be the same as blocks

db file parallel write

This event occurs in the DBWR. It indicates that the DBWR is performing a parallel write to files and blocks. When the last I/O has gone to disk, the wait ends.

Wait Time: Wait until all of the I/Os are completed

Parameter	Description
requests	This indicates the total number of I/O requests, which will be the same as blocks
interrupt	
timeout	This indicates the timeout value in centiseconds to wait for the IO completion.

db file scattered read

Similar to **db file sequential read**, except that the session is reading multiple data blocks.

Wait Time: The wait time is the actual time it takes to do all of the I/Os

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
blocks	The number of blocks that the session is trying to read from the *file#* starting at *block#*

db file sequential read

The session waits while a sequential read from the database is performed. This event is also used for rebuilding the control file, dumping datafile headers, and getting the database file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
blocks	This is the number of blocks that the session is trying to read (should be 1)

db file single write

This event is used to wait for the writing of the file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
blocks	This is the number of blocks that the session is trying to write in *file# starting at block#*

DFS db file lock

This event occurs only for the DBWR in the Oracle Real Application Cluster. Each DBWR of every instance holds a global lock on each file in shared mode. The instance that is trying to offline the file will escalate the global lock from shared to exclusive. This signals the other instances to synchronize their SGAs with the control file before the file can be taken offline. The name of this lock is **DF** (see Appendix D, "Oracle Enqueue Names" for more information).

Wait Time: 1 second in loop. The DBWR is waiting in a loop (sleep, check) for the other instances to downgrade to NULL mode. During this time, the DBWR cannot perform other tasks such as writing buffers.

Parameter	Description
file	See "file#" on page C-3

DFS lock handle

The session waits for the lock handle of a global lock request. The lock handle identifies a global lock. With this lock handle, other operations can be performed on this global lock (to identify the global lock in future operations such as conversions or release). The global lock is maintained by the DLM.

Wait Time: The session waits in a loop until it has obtained the lock handle from the DLM. Inside the loop there is a wait of 0.5 seconds.

Parameter	Description
name	See "name and type" on page C-4
mode	See "mode" on page C-3
id1	See "id1" on page C-3
id2	See "id2" on page C-3

The session needs to get the lock handle.

direct path read

During Direct Path operations the data is asynchronously read from the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if during a direct read no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	Number of valid buffers in the context referenced by the descriptor address

direct path write

During Direct Path operations, the data is asynchronously written to the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if, during a direct write, no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	Number of valid buffers in the context referenced by the descriptor address

dispatcher shutdown

During shutdown immediate or normal, the shutdown process must wait for all the dispatchers to shutdown. As each dispatcher is signaled, the session that causes the shutdown is waits on this event until the requested dispatcher is no longer alive.

Wait Time: 1 second

Parameter	Description
waited	Indicates the cumulative wait time. After 5 minutes, the session writes to the alert and trace files to indicate that there might be a problem.

dispatcher timer

This basically means that the dispatcher is idle and waiting for some work to arrive.

Wait Time: 60 seconds

Parameter	Description
sleep time	The intended sleep time. The dispatcher will return to work sooner if it is posted by either data arriving on the network or by a post from a shared server process to send data back to the client.

duplicate cluster key

It is possible for a race condition to occur when creating a new cluster key. If it is found that another process has put the cluster key into the data/index block, then the session waits and retries. The retry should then find a valid cluster key.

Wait Time: 0.01 seconds

Parameter	Description
dba	The dba of the block into which the session is trying to insert a cluster key

enq: OW - initialization

A session will wait on this event if it is trying to initialize the database wallet, and another session has already begun an initialization.

Wait Time: Total time necessary to initialize the wallet context

Parameters: None

enq: OW - termination

A session will wait on this event if it is trying to terminate the database wallet, and another session has already begun a termination.

Wait Time: Total time necessary to deallocate memory used by the wallet context and terminate the context.

Parameters: None

enqueue

The session is waiting for a local enqueue. The wait is dependent on the name of the enqueue (see Appendix D, "Oracle Enqueue Names").

Wait Time: Depends on the enqueue name

Parameter	Description
name	See "name and type" on page C-4
mode	See "mode" on page C-3

flashback buf free by RVWR

This wait event only occurs when Flashback Database is turned on. A session waits for recovery writer (RVWR) to write flashback data to the flashback logs on disk because the buffers are full. Until RVWR can free up the buffers, the session may need to wait.

If this event becomes a top wait event for the database, it is typically because the file system or storage system for the Flash Recovery Area does not support enough bandwidth for Oracle to write the flashback database logs. Refer to the Flashback Database section in *Oracle Database Backup and Recovery Basics* for tuning considerations.

Wait Time: 1 second

Parameters: None

free buffer waits

This will happen if:

- All buffer gets have been suspended. This could happen when a file was read-only and is now read-write. All the existing buffers need to be invalidated since they are not linked to lock elements (needed when mounted parallel (shared)). So cache buffers are not assigned to data block addresses until the invalidation is finished.
- The session moved some dirty buffers to the dirty queue and now this dirty queue is full. The dirty queue needs to be written first. The session will wait on this event and try again to find a free buffer
- This also happens after inspecting **free buffer inspected** buffers. If no free buffer is found, Oracle waits for one second, and then tries to get the buffer again (depends on the context). For more information, see free buffer inspected on page E-6.

Wait Time: 1 second

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2

free global transaction table entry

The session is waiting for a free slot in the global transaction table (used by the Distributed Database option). It will wait for 1 second and try again.

Wait Time: 1 second

Parameter	Description
tries	The number of times the session tried to find a free slot in the global transaction table

free process state object

Used during the creation of a process. The session will scan the process table and look for a free process slot. If none can be found, PMON is posted to check if all the processes currently in the process table are still alive. If there are dead processes, then PMON will clean them and make the process slot available to new processes. The waiting process will then rescan the process table to find the new slot.

Wait Time: 1 second

Parameters: None

GCS lock open S

The session waits for a resource get in `SHARED` mode on the block identified by file# and block#.

Wait Time: 1 second

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
class	See "class" on page C-3

GCS lock open X

The session waits for a resource get in `EXCLUSIVE` mode on the block identified by file# and block#.

Wait Time: 1 second

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
lenum	See "le" on page C-3

gcs remastering wait for drop pkey

A session dropping an object waits on the lock manager daemon (LMD) to remove the object's affinity to an instance.

Wait Time: 20 centi-seconds

Parameter	Description
pkey	The object id of the database object being dropped

global cache busy

The session waits to convert a buffer from Shared Current to Exclusive Current status.

Wait Time: 1 second

Parameter	Description
file#	See "file#" on page C-3

Parameter	Description
block#	See "block#" on page C-2
le	See "lenum" on page A-8

global cache lock cleanup

PMON is waiting for an LCK process to cleanup the lock context after a foreground process died while doing a global cache lock operation.

Wait Time: 1 second

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
lenum	See "lenum" on page A-8

global cache freelist

All releasable locks are used and a new one has been requested. To make a resource element available, a resource element is pinged.

Wait Time: The duration of the resource get operation to ping the resource element

Parameters: None

inactive session

This event is used for two purposes:

- Switching sessions

  If a time-out period has been specified, then wait that amount of time for the session to be detached.

- Killing sessions

  From either `KILL SESSION` or internal request. Having posted a session that it should kill itself, wait for up to 1 minute for the session to terminate.

Wait Time: 1 second

Parameter	Description
session#	See "session#" on page C-4
waited	See "waited" on page C-4

inactive transaction branch

The session waits for a transaction branch that is currently used by another session.

Wait Time: 1 second

Parameter	Description
branch#	The serial number of the transaction for which the session is waiting
waited	See "waited" on page C-4

index block split

While trying to find an index key in an index block, Oracle noticed that the index block was being split. Oracle will wait for the split to finish and try to find the key again.

Wait Time: The session will yield the CPU, so there is no actual waiting time

Parameter	Description
rootdba	The root of the index
level	This is the level of the block that the session is trying to split in the index. The leaf blocks are level 0. If the level is > 0, it is a branch block. (The root block can be considered a special branch block).
childdba	The block that the session is trying to split

instance state change

The session waits for SMON to enable or disable cache or transaction recovery. This usually happens during `ALTER DATABASE OPEN` or `CLOSE`.

Wait Time: Wait time depends on the amount of time the action takes (that is, the amount of recovery needed)

Parameter	Description
layer	This value can be 1 or 2. If 1, it means that the transaction layer wants transaction recovery to be performed. If 2, it means that cache recovery will be performed.
value	This value can be 0 (disable) or 1 (enable)
waited	The number of seconds waited so far

io done

The session waits for an I/O to complete or it waits for a slave process to become available to submit the I/O request. This event occurs on platforms that do not support asynchronous I/O.

Wait Time: 50 milliseconds

Parameter	Description
msg ptr	A pointer to the I/O request

kcl bg acks

The session waits for the background LCK process(es) to finish what they are doing. For example:

- Lock recovery
- Initializing the locks (start up)
- Finalizing the locks (shut down)

Wait Time: 10 seconds

Parameter	Description
count	The number of LCK processes that have finished

Parameter	Description
loops	The number times the process had to wait for the LCK processes to finish what they were doing

ksxr wait for mount shared

The cross instance broadcast facility of this Oracle instance is waiting for the database mount in shared mode to complete.

Wait Time: The time taken for the instance to mount. An indefinite wait on this event implies that the instance startup is hanging.

ktm: instance recovery

The session waits for SMON to finish the instance, transaction recovery, or sort segment cleanup.

Wait Time: The wait time can vary and depends on the amount of recovery needed

Parameter	Description

undo segment#	If the value is 0, SMON is probably performing instance recovery. If P1 > 0, use this query to find the undo segment:
	`select *`
	`from v$rollstat`
	`where usn = undo segment#;`

latch activity

This event is used as part of the process of determining whether a latch needs to be cleaned.

Wait Time: 0.05 to 0.1 seconds

Parameter	Description
address	The address of the latch that is being checked

number	The latch number of the latch that has activity. To find more information on the latch, use the following SQL statement:
	`select *`
	`from v$latchname`
	`where latch# = number;`

process#	If this is 0, it is the first phase of the in-flux tests

latch free

The process waits for a latch that is currently busy (held by another process).

Wait Time: The wait time increases exponentially and does not include spinning on the latch (active waiting). The maximum wait time also depends on the number of latches that the process is holding. There is an incremental wait of up to 2 seconds.

Parameter	Description
address	The address of the latch for which the process is waiting

Parameter	Description

number	The latch number that indexes in the `V$LATCHNAME` view.To find more information on the latch, use the following SQL statement:
	`select *`
	`from v$latchname`
	`where latch# = number;`

tries	A count of the number of times the process tried to get the latch (slow with spinning) and the process has to sleep

library cache load lock

The session tries to find the load lock for the database object so that it can load the object. The load lock is always obtained in Exclusive mode, so that no other process can load the same object. If the load lock is busy the session will wait on this event until the lock becomes available.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
object address	Address of the object being loaded
lock address	Address of load lock being used
mask	Indicates which data pieces of the object that needs to be loaded

library cache lock

This event controls the concurrency between clients of the library cache. It acquires a lock on the object handle so that either:

- One client can prevent other clients from accessing the same object
- The client can maintain a dependency for a long time (for example, no other client can change the object)

This lock is also obtained to locate an object in the library cache.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
handle address	Address of the object being loaded
lock address	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is a State Object.
mode	Indicates the data pieces of the object which need to be loaded
namespace	See "namespace" on page C-4

library cache pin

This event manages library cache concurrency. Pinning an object causes the heaps to be loaded into memory. If a client wants to modify or examine the object, the client must acquire a pin after the lock.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
handle address	Address of the object being loaded
pin address	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is basically a State Object.
mode	Indicates which data pieces of the object that needs to be loaded
namespace	See "namespace" on page C-4

library cache shutdown

The process shutting down the instance waits for sessions to complete before proceeding with library cache shutdown.

LMON global data update

The rolling migration operation is waiting for a response from LMON to acknowledge the global data was updated.

Wait Time: The time it takes for LMON to publish/retrieve the global data associated with a

Parameters: None

lock manager wait for remote message

The lock manager waits for a message from a remote lock manager in the same configuration.

Wait Time: The elapsed time of the wait

Parameter	Description
waittime	The elapsed time of the actual wait

log buffer space

Waiting for space in the log buffer because the session is writing data into the log buffer faster than LGWR can write it out. Consider making the log buffer bigger if it is small, or moving the log files to faster disks such as striped disks.

Wait Time: Usually 1 second, but 5 seconds if it is waiting for a Switch Logfile to complete

Parameters: None

log file parallel write

Writing redo records to the redo log files from the log buffer.

Wait Time: Time it takes for the I/Os to complete. Even though redo records are written in parallel, the parallel write is not complete until the last I/O is on disk.

Parameter	Description
files	Number of files to be written
blocks	Number of blocks to be written
requests	Number of I/O requests

log file sequential read

Waiting for the read from this logfile to return. This is used to read redo records from the log file.

Wait Time: Time it takes to complete the physical I/O (read)

Parameter	Description
log#	The relative sequence number of the logfiles within a log group (used only when dumping the logfiles)
block#	See "block#" on page C-2
blocks	The number of blocks to read

log file single write

Waiting for the write to this logfile to complete. This event is used while updating the header of the logfile. It is signaled when adding a log file member and when incrementing sequence numbers.

Wait Time: Time it takes for the physical I/O (write) to complete

Parameter	Description
log#	This is the number of the group/log to which the session is currently writing
block#	See "block#" on page C-2
blocks	The number of blocks to write

log file switch (archiving needed)

Waiting for a log switch because the log that the LGWR will be switching into has not been archived yet.   Check the alert file to make sure that archiving has not stopped due to a failed archive write. To speed archiving, consider adding more archive processes or putting the archive files on striped disks.

Wait Time: 1 second

Parameters: None

log file switch (checkpoint incomplete)

Waiting for a log switch because the session cannot wrap into the next log. Wrapping cannot be performed because the checkpoint for that log has not completed.

Wait Time: 1 second

Parameters: None

log file switch (clearing log file)

Waiting for a log switch because the log is being cleared due to a `CLEAR LOGFILE` command or implicit clear logfile executed by recovery.

Wait Time: 1 second

Parameters: None

log file switch (private strand flush incomplete)

User sessions trying to generate redo, wait on this event when LGWR waits for DBWR to complete flushing redo from IMU buffers into the log buffer; when DBWR is complete LGWR can then finish writing the current log, and then switch log files.

Wait Time: 1 second

Parameters: None

log file switch completion

Waiting for a log switch to complete.

Wait Time: 1 second

Parameters: None

log file sync

When a user session commits, the session's redo information needs to be flushed to the redo logfile. The user session will post the LGWR to write the log buffer to the redo log file. When the LGWR has finished writing, it will post the user session.

Wait Time: The wait time includes the writing of the log buffer and the post.

Parameter	Description
buffer#	The number of the physical buffer in the redo log buffer that needs to be synchronized

log switch/archive

Used as part of the `ALTER SYSTEM ARCHIVE LOG CHANGE` *scn* statement. The session waits for the current log from all open threads to be archived.

Wait Time: Wait for up to 10 seconds

Parameter	Description
thread#	The thread number of the thread that is currently archiving its current log

optimizer stats update retry

When concurrent sessions try to update optimizer statistics for the same object, all of them except the one that successfully acquired all necessary locks/pins on the library/row cache entries, wait on this wait event and then retry locking after a short period of time. In addition to explicit statistics gathering and maintenance operations using the DBMS_STATS package, Oracle database itself may try to update statistics for some objects, either on behalf of the user or for its own maintenance purposes.

Wait Time: 10 ms

Parameters: None

parallel recovery change buffer free

The parallel recovery coordinator is waiting for a change mapping buffer to be released by one of the recovery slaves.

Wait Time: 100ms

Parameters: None

parallel recovery control message reply

The parallel recovery coordinator is waiting for all recovery slaves to respond to a synchronous control message.

Wait Time: 100ms

Parameters: None

parallel recovery coord send blocked

The parallel recovery coordinator is unable to send a redo change message because the recovery slave is still actively applying redo that it has already received and has not yet released the channel.

Wait Time: 100ms

Parameters: None

parallel recovery coord wait for reply

The parallel recovery coordinator is waiting for all recovery slaves to exit.

Wait Time: 100ms

Parameters: None

parallel recovery coordinator waits for slave cleanup

The parallel recovery coordinator is waiting for all recovery slaves to exit gracefully.

Wait Time: 10ms

Parameters: None

parallel recovery read buffer free

The parallel recovery coordinator is waiting for a log read buffer to be released by all recovery slaves. Only after every recovery slave finishes applying redo from a log read buffer, can the buffer be used by the coordinator to issue the next log read.

Wait Time: 100ms

Parameters: None

parallel recovery slave next change

A parallel recovery slave is idle and waiting for the next change message from the coordinator.

Wait Time: 100ms

Parameters: None

pending global transaction(s)

This event should happen only during testing. The session waits for pending transactions to clear.

Wait Time: 30 seconds

Parameter	Description
scans	Number of times the session has scanned the PENDING_TRANS$ table

pipe get

The session waits for a message to be received on the pipe or for the pipe timer to expire.

Wait Time: There is a 5 second wake up (check) and the pipe timer set by the user

Parameter	Description
handle address	The library cache object handle for this pipe
buffer length	The length of the buffer
timeout	The pipe timer set by the user

pipe put

The session waits for the pipe send timer to expire or for space to be made available in the pipe.

Wait Time: There is the 5 second wakeup (check) and the user-supplied timeout value

Parameter	Description
handle address	The library cache object handle for this pipe
record length	The length of the record or buffer that has been put into the pipe
timeout	The pipe timer set by the user

PL/SQL lock timer

This event is called through the DBMS_LOCK.SLEEP procedure or USER_LOCK.SLEEP procedure. This event will most likely originate from procedures written by a user.

Wait Time: The wait time is in hundredths of seconds and is dependent on the user context

Parameter	Description
duration	The duration that the user specified in the DBMS_LOCK.SLEEP or USER_LOCK.SLEEP procedures

pmon timer

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

Wait Time: Up to 3 seconds, if not posted before

Parameter	Description
duration	The actual amount of time that the PMON is trying to sleep

prewarm transfer retry

Release a hash latch, then wait under this event before attempting to re-acquire the hash latch.

Wait Time: 10ms

Parameters: None

prior process spawner to be cleaned up

When a prior process has died while spawning a background, the current process which is trying to spawn new a background must wait until the prior process state is cleaned up.

Wait Time: Usually 3 - 10 seconds

Parameter	Description
process_pid	process identifier (see `V$PROCESS.PID`) of the process whose state needs to be cleaned up.
process_sno	process serial number (see `V$PROCESS.SERIAL#`) of the process whose state needs to be cleaned up.

process startup

Wait for a shared server, Dispatcher, or other background process to start.

Wait Time: Wait up to 1 second for a background process to start. If timed out, then re-wait until 5 minutes have passed and signal an error. If the process has started, the event will acknowledge this.

Parameter	Description
type	The process type that was started
process#	The process number of the process being started
waited	Cumulative time waited for the process to start

PX Deque wait

The process is waiting for a message during a parallel execute.

Wait Time: The wait time depends on how quickly the message arrives. Wait times can vary, but it will normally be a short period of time.

Parameter	Description
reason	The reason for dequeuing
sleeptime	The amount of time that the session slept
loop	The total number of times that the session has slept

PX qref latch

Each parallel execution process has a parallel execution qref latch, which needs to be acquired before the queue buffers can be manipulated.

Wait Time: Wait up to 1 second

Parameter	Description
function	Indicates the type of wait that the session is doing
sleeptime	The amount of time that the session waits (in hundredths of a second)
qref	The address of the process queue for which the session is waits

PX server shutdown

During normal or immediate shutdown the parallel execution slaves are posted to shutdown cleanly. If any parallel execution slaves are still alive after 10 seconds, they are killed.

Wait Time: Wait up to 0.5 seconds

Parameter	Description
nalive	The number of parallel execution slaves that are still running
sleeptime	The total sleeptime since the session started to wait on this event
loop	The number of times the session waited for this event

PX signal server

This event occurs only in Exclusive mode. The query coordinator is signalling the Query Slaves that an error has occurred.

Wait Time: 0.5 seconds

Parameter	Description
serial	The serial number of the slave process queue
error	The error that has occurred
nbusy	The number of slave processes that are still busy

Streams AQ: waiting for messages in the queue

The session is waiting on an empty OLTP queue (Advanced Queuing) for a message to arrive so that the session can dequeue that message.

Wait Time: The amount of time that the session wants to wait is determined by the parameter *wait time*

Parameter	Description
queue id	The ID of the OLTP queue for which this session is waiting
process#	The process number of the process in which this session runs
wait time	The intended wait time for this session

rdbms ipc message

The background processes (LGWR, DBWR, LMS0) use this event to indicate that they are idle and are waiting for the foreground processes to send them an IPC message to do some work.

Wait Time: Up to 3 seconds. The parameter *timeout* shows the true sleep time.

Parameter	Description
timeout	The amount of time that the session waits for an IPC message

rdbms ipc message block

This event indicates that all message blocks are in use and that the session had to wait for a message block to become available.

Wait Time: Wait up to 60 seconds

Parameters: None

rdbms ipc reply

This event is used to wait for a reply from one of the background processes.

Wait Time: The wait time is specified by the user and is indicated by the parameter *timeout*.

Parameter	Description
from_process	The background process for which the session is waiting. The wait is for a reply to an IPC message sent by the session.
timeout	The amount of time in seconds that this process will wait for a reply

read by other session

This event occurs when a session requests a buffer that is currently being read into the buffer cache by another session. Prior to release 10.1, waits for this event were grouped with the other reasons for waiting for buffers under the 'buffer busy wait' event

Wait Time: Time waited for the buffer to be read by the other session (in microseconds)

Parameter	Description
file#	See "file#" on page C-3
block#	See "block#" on page C-2
class#	See "class" on page C-3

recovery read

A parallel recovery slave (or serial recovery process) is waiting for a batch of data block reads to complete.

Wait Time: Time it takes to complete the physical I/O (read)

Parameters: None

resmgr: become active

The session is waiting for a resource manager active session slot. This event occurs when the resource manager is enabled and the number of active sessions in the session’s current consumer group exceeds the current resource plan’s active session limit for the consumer group. To reduce the occurrence of this wait event, increase the active session limit for the session’s current consumer group.

Wait Time: The time the session waited to be allocated an active session slot

Parameter	Description
location	location of the wait

resmgr: cpu quantum

The session is waiting to be allocated a quantum of cpu. This event occurs when the resource manager is enabled and is throttling CPU consumption. To reduce the occurrence of this wait event, increase the CPU allocation for the sessions's current consumer group.

Wait Time: The time the session waited to acquire a CPU quantum

Parameter	Description
location	Location of the wait

rolling migration: cluster quisce

This is the wait event that instances wait on when cluster is about to start a rolling migration. The instances are waiting for any privileged operations that blocks rolling migration to complete before allowing rolling migration.

Wait Time: 1 second

Parameter	Description
location	Its value will be 1 if the wait is for completion of the privileged operations so that a rolling upgrade/downgrade can start. Its value will be 2 if the wait is for completion of the rolling upgrade/downgrade on all the nodes in the cluster.
waits	The number of seconds spent waiting at the current location.

row cache lock

The session is trying to get a data dictionary lock.

Wait Time: Wait up to 60 seconds.

Parameter	Description
cache id	The `CACHE#` column value in the `V$ROWCACHE` view
mode	See "mode" on page C-3
request	The pipe timer set by the user

sbtbufinfo

This function is called when Oracle needs to discover the size, and number, of I/O buffers that have been allocated by the SBT layer. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

sbtgetbuf

This function obtains one I/O buffer that Oracle will use for I/O during a backup job.

Wait Time: Less than one millisecond

Parameters: None

sbtmapbuf

This is an internal function used to facilitate multi-process buffer management. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

sbtrelbuf

This function releases an I/O buffer that has been already processed during a restore job, so that the SBT layer can fill it with more data. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

scginq AST call

Called by the session to find the highest lock mode that is held on a resource.

Wait Time: Wait up to 0.2 seconds, but the wait will continue until the NULL mode Acquisition AST has fired.

Parameters: None

SGA: allocation forcing component growth

Process waiting on an immediate mode memory transfer with auto-tune SGA after a 4031 for MMAN to get the memory and post it.

Wait Time: 10 msec

Parameters: None

SGA: MMAN sleep for component shrink

MMAN to wait and post itself for satisfying an auto-tuned memory request while trying to fully free a component's quiesced granules. In Release 10.1, the name of this event was 'wait for SGA component shrink'.

Wait Time: 10 msec

Parameter	Description
P1	component_id (corresponding to the memory pool)
P2	Current size in granules
P3	Target size in granules

SGA: sga_target resize

Memory resize requests wait while sga target is being resized.In Release 10.1, the name of this event was 'wait for sga_target resize'.

Wait Time: 10 msec

Parameters: None

Shared IO Pool Memory

Wait until a shared I/O pool buffer becomes available. This happens when processes are using these buffers for I/O and the current process needs to wait for the release of any one of the buffers to the shared I/O pool.

Wait Time: 10msec

Parameters: None

single-task message

When running single task, this event indicates that the session waits for the client side of the executable.

Wait Time: Total elapsed time that this session spent in the user application

Parameters: None

smon timer

This is the main idle event for SMON. SMON will be waiting on this event most of the time until it times out or is posted by another process.

Wait Time: 5 minutes (300 seconds)

Parameter	Description
sleeptime	The amount of time that SMON tries to wait on this event in seconds
failed	The number of times SMON was posted when there some kind of error

SQL*Net break/reset to client

The server sends a break or reset message to the client. The session running on the server waits for a reply from the client.

Wait Time: The actual time it takes for the break or reset message to return from the client

Parameter	Description
driver id	See "driver id" on page C-3
break?	See "break?" on page C-3

SQL*Net break/reset to dblink

Same as **SQL*Net break/reset to client**, but in this case, the break/reset message is sent to another server process over a database link.

Wait Time: The actual time it takes for the break or reset message to return from the other server process

Parameter	Description
driver id	See "driver id" on page C-3
break?	See "break?" on page C-3

SQL*Net message from client

The server process (foreground process) waits for a message from the client process to arrive.

Wait Time: The time it took for a message to arrive from the client since the last message was sent to the client

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes received by the server (foreground process) from the client.

SQL*Net message from dblink

The session waits while the server process (foreground process) receives messages over a database link from another server process.

Wait Time: The time it took for a message to arrive from another server (foreground process) since a message was sent to the other foreground process.

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes received by the server (foreground process) from another foreground process over a database link.

SQL*Net message to client

The server (foreground process) is sending a message to the client.

Wait Time: The actual time the **send** takes

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes sent by the server process to the client

SQL*Net message to dblink

The server process (foreground process) is sending a message over a database link to another server process.

Wait Time: The actual time the **send** takes

Parameter	Description
driver id	See "driver id" on page C-3

Parameter	Description
#bytes	The number of bytes sent by the server process to another server process over a database link

SQL*Net more data from client

The server is waiting on the client to send more data to its client shadow process, in an already initiated operation.

Wait Time: The time waited depends on the time it took to receive the data (including the waiting time)

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes received from the client

SQL*Net more data from dblink

The foreground process is expecting more data from a data base link.

Wait Time: The total time it takes to read the data from the database link (including the waiting time for the data to arrive)

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes received

SQL*Net more data to client

The server process is sending more data/messages to the client. The previous operation to the client was also a **send**.

Wait Time: The actual time it took for the **send** to complete

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes that are being sent to the client

SQL*Net more data to dblink

The event indicates that the server is sending data over a database link again. The previous operation over this database link was also a **send**.

Wait Time: The actual time it takes to send the data to the other server

Parameter	Description
driver id	See "driver id" on page C-3
#bytes	The number of bytes that are sent over the database link to the other server process

switch logfile command

The session waits on the user command `SWITCH LOGFILE` to complete.

Wait Time: 5 seconds

Parameters: None

timer in sksawat

The session waits for the Archiver (ARCH) asynchronous I/O to complete.

Wait Time: 0.01 seconds

Parameters: None

transaction

Wait for a blocking transaction to be rolled back. Continue waiting until the transaction has been rolled back.

Wait Time: 1 second

Parameter	Description
undo seg#	The rollback segment ID
slot#	The slot ID inside the rollback segment
wrap#	The sequence number that is incremented for each transaction
count	The number of times that the session has waited on this transaction

unbound tx

The session waits to see if there are any transactions that have been started but do not have a Rollback Segment associated with them.

Wait Time: 1 second

Parameters: None

undo_retention publish retry

This wait can occur for two reasons. A session issuing an `ALTER SYSTEM SET UNDO_RETENTION` may wait on this event wait while a cluster reconfiguration takes place. Or the background process MMNL may wait for cluster reconfiguration while attempting to determine the max `UNDO_RETENTION`.

Wait time: 1 second

Parameter	Description
P1	Identifies where the retry is happening. Id = 1 retry while publishing into the max undo_retention namespace. Id = 2 retry while iterator accessing the max undo_retention namespace
P2	Retry count (maximum number of retries is 5)

undo segment extension

The undo segment is being extended or shrunk. The session must wait until the operation on the undo segment has finished.

Wait Time: 0.01 seconds

Parameter	Description
segment#	The ID of the rollback segment that is being extended or shrunk

undo segment recovery

PMON is rolling back a dead transaction. The wait continues until rollback finishes.

Wait Time: 3 seconds

Parameter	Description
segment#	The ID of the rollback segment that contains the transaction that is being rolled back
tx flags	The transaction flags (options) set for the transaction that is being rolled back

undo segment tx slot

Wait for a transaction slot to become available within the selected rollback segment. Continue waiting until the slot is available.

Wait Time: 1 second

Parameter	Description
segment#	The ID of the rollback segment that contains the transaction that is being rolled back

virtual circuit status

The session waits for a virtual circuit to return a message type indicated by *status*.

Wait Time: 30 seconds

Parameter	Description
circuit#	Indicates the virtual circuit# being waited on
status	Indicates what the session is waiting for

WMON goes to sleep

WMON is the UNIX-specific Wait Monitor, that can be used to reduce the number of system calls related to setting timers for posting or waiting in Oracle. You need to set an initialization parameter that enables the WMON process.

Wait Time: Depends on the next timeout

Parameters: None

write complete waits

The session waits for a buffer to be written. The write is caused by normal aging or by a cross-instance call.

Wait Time: 1 second

Parameter	Description
file#	The rollback segment id that contains the transaction that is being rolled back
block#	The transaction flags (options) set for the transaction that is being rolled back
id	Identifies the reason for waiting

writes stopped by instance recovery or database suspension

The session is blocked until the instance that started Instance Recovery is finished.

Wait Time: 5 seconds

Parameter	Description
bythread#	The rollback segment id that contains the transaction that is being rolled back
ourthread#	The current instance thread number

D

Oracle Enqueue Names

This appendix lists Oracle enqueues. **Enqueues** are shared memory structures (locks) that serialize access to database resources. They can be associated with a session or transaction. Enqueue names are displayed in the `LOCK_TYPE` column of the `DBA_LOCK` and `DBA_LOCK_INTERNAL` data dictionary views.

A **resource** uniquely identifies an object that can be locked by different sessions within an instance (local resource) or between instances (global resource). Each session that tries to lock the resource will have an **enqueue** on the resource.

> **Note:** The names of enqueues and their definitions may change from release to release.

See Also: "DBA_LOCK_INTERNAL" on page 4-142 and "DBA_LOCK" on page 4-141

The Oracle enqueues are:

- BL, Buffer Cache Management
- BR, Backup/Restore
- CF, Controlfile Transaction
- CI, Cross-instance Call Invocation
- CU, Bind Enqueue
- DF, Datafile
- DL, Direct Loader Index Creation
- DM, Database Mount
- DR, Distributed Recovery Process
- DX, Distributed Transaction
- FP, File Object
- FS, File Set
- HW, High-Water Lock
- IN, Instance Number
- IR, Instance Recovery
- IS, Instance State

- IV, Library Cache Invalidation
- JI, Enqueue used during AJV snapshot refresh
- JQ, Job Queue
- KK, Redo Log "Kick"
- KO, Multiple Object Checkpoint
- L[A-P], Library Cache Lock
- LS, Log Start or Switch
- MM, Mount Definition
- MR, Media Recovery
- N[A-Z], Library Cache Pin
- PE, ALTER SYSTEM SET PARAMETER = VALUE
- PF, Password File
- PI, Parallel Slaves
- PR, Process Startup
- PS, Parallel Slave Synchronization
- Q[A-Z], Row Cache
- RO, Object Reuse
- RT, Redo Thread
- RW, Row Wait
- SC, System Commit Number
- SM, SMON
- SN, Sequence Number
- SQ, Sequence Number Enqueue
- SR, Synchronized Replication
- SS, Sort Segment
- ST, Space Management Transaction
- SV, Sequence Number Value
- TA, Transaction Recovery
- TC, Thread Checkpoint
- TE, Extend Table
- TM, DML Enqueue
- TO, Temporary Table Object Enqueue
- TS, Temporary Segment (also TableSpace)
- TT, Temporary Table
- TX, Transaction
- UL, User-defined Locks
- UN, User Name

- US, Undo Segment, Serialization
- WL, Being Written Redo Log
- XA, Instance Attribute Lock
- XI, Instance Registration Lock

E

Statistics Descriptions

This appendix describes the statistics stored in the V$SESSTAT and V$SYSSTAT dynamic performance tables. These statistics are useful in identifying and correcting performance problems.

This appendix contains the following topics:

- Displaying Statistics
- Statistics Descriptions

Displaying Statistics

The V$SESSTAT view displays statistics on a per-session basis and is valid only for the session currently connected. When a session disconnects, all statistics for the session are updated in V$SYSSTAT. The values for the statistics are cleared until the next session uses them.

The V$STATNAME view contains all of the statistics for an Oracle release.

Many of these statistics are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice, even between patch releases. Application developers should be aware of this and write their code to tolerate missing or extra statistics.

> **See Also:** "V$SESSTAT" on page 8-20, "V$STATNAME" on page 8-55, and "V$SYSSTAT" on page 8-67 for more information on these views

Statistics Descriptions

This section describes the statistics stored in the V$SESSTAT and V$SYSSTAT views. The statistics are listed here in alphabetical order.

The CLASS column contains a number representing one or more statistics classes. The following class numbers are additive:

- 1, User
- 2, Redo
- 4, Enqueue
- 8, Cache

- 16, OS
- 32, Real Application Clusters
- 64, SQL
- 128, Debug

For example, a class value of 72 represents a statistic that relates to SQL statements and caching.

Some statistics are populated only if the TIMED_STATISTICS initialization parameter is set to true. Those statistics are flagged in the right-hand column.

Table E–1 Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
background checkpoints completed	8	Number of checkpoints completed by the background process. This statistic is incremented when the background process successfully advances the thread checkpoint.	
background checkpoints started	8	Number of checkpoints started by the background process. This statistic can be larger than "background checkpoints completed" if a new checkpoint overrides an incomplete checkpoint or if a checkpoint is currently under way. This statistic includes only checkpoints of the redo thread. It does not include: ■ Individual file checkpoints for operations such as offline or begin backup ■ Foreground (user-requested) checkpoints (for example, performed by ALTER SYSTEM CHECKPOINT LOCAL statements)	
background timeouts	128	This is a count of the times where a background process has set an alarm for itself and the alarm has timed out rather than the background process being posted by another process to do some work.	
branch node splits	128	Number of times an index branch block was split because of the insertion of an additional value	
buffer is not pinned count	72	Number of times a buffer was free when visited. Useful only for internal debugging purposes.	
buffer is pinned count	72	Number of times a buffer was pinned when visited. Useful only for internal debugging purposes.	
bytes received via SQL*Net from client	1	Total number of bytes received from the client over Oracle Net Services	
bytes received via SQL*Net from dblink	1	Total number of bytes received from a database link over Oracle Net Services	
bytes sent via SQL*Net to client	1	Total number of bytes sent to the client from the foreground processes	
bytes sent via SQL*Net to dblink	1	Total number of bytes sent over a database link	
Cached Commit SCN referenced	128	Useful only for internal debugging purposes	
calls to get snapshot scn: kcmgss	32	Number of times a snapshot system change number (SCN) was allocated. The SCN is allocated at the start of a transaction.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
calls to kcmgas	128	Number of calls to routine kcmgas to get a new SCN	
calls to kcmgcs	128	Number of calls to routine kcmgcs to get a current SCN	
calls to kcmgrs	128	Number of calls to routine kcsgrs to get a recent SCN	
change write time	8	Elapsed redo write time for changes made to `CURRENT` blocks in 10s of milliseconds.	3
cleanouts and rollbacks - consistent read gets	128	Number of consistent gets that require both block rollbacks and block cleanouts. **See Also:** "consistent gets"	
cleanouts only - consistent read gets	128	Number of consistent gets that require only block cleanouts, no rollbacks. **See Also:** "consistent gets"	
cluster key scan block gets	64	Number of blocks obtained in a cluster scan	
cluster key scans	64	Number of cluster scans that were started	
cold recycle reads	8	Number of buffers that were read through the least recently used end of the recycle cache with fast aging strategy	
commit cleanout failures: block lost	8	Number of times Oracle attempted a cleanout at commit but could not find the correct block due to forced write, replacement, or switch `CURRENT`	
commit cleanout failures: buffer being written	8	Number of times Oracle attempted a cleanout at commit, but the buffer was currently being written	
commit cleanout failures: callback failure	8	Number of times the cleanout callback function returns `FALSE`	
commit cleanout failures: cannot pin	8	Total number of times a commit cleanout was performed but failed because the block could not be pinned	
commit cleanout failures: hot backup in progress	8	Number of times Oracle attempted block cleanout at commit during hot backup. The image of the block needs to be logged before the buffer can be made dirty.	
commit cleanout failures: write disabled	8	Number of times a cleanout block at commit was performed but the writes to the database had been temporarily disabled	
commit cleanouts	8	Total number of times the cleanout block at commit function was performed	
commit cleanouts successfully completed	8	Number of times the cleanout block at commit function completed successfully	
Commit SCN cached	128	Number of times the system change number of a commit operation was cached	
consistent changes	8	Number of times a user process has applied rollback entries to perform a consistent read on the block Work loads that produce a great deal of consistent changes can consume a great deal of resources. The value of this statistic should be small in relation to the "consistent gets" statistic.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
consistent gets	8	Number of times a consistent read was requested for a block. **See Also:** "consistent changes" and "session logical reads" statistics	
consistent gets direct	8	Number of times a consistent read was requested for a block bypassing the buffer cache (for example, direct load operation). This is a subset of "consistent gets" statistics value.	
consistent gets from cache	8	Number of times a consistent read was requested for a block from buffer cache. This is a subset of "consistent gets" statistics value.	
CPU used by this session	1	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added. A similar problem can exist in the reporting by the operating system, especially on systems that suffer from many context switches.	3
CPU used when call started	128	The CPU time used when the call is started **See Also:** "CPU used by this session"	3
CR blocks created	8	Number of `CURRENT` blocks cloned to create CR (consistent read) blocks. The most common reason for cloning is that the buffer is held in a incompatible mode.	
current blocks converted for CR	8	Number `CURRENT` blocks converted to CR state	
cursor authentications	128	Number of privilege checks conducted during execution of an operation	
data blocks consistent reads - undo records applied	128	Number of undo records applied to data blocks that have been rolled back for consistent read purposes	
db block changes	8	Closely related to "consistent changes", this statistic counts the total number of changes that were part of an update or delete operation that were made to all blocks in the SGA. Such changes generate redo log entries and hence become permanent changes to the database if the transaction is committed. This approximates total database work. It statistic indicates the rate at which buffers are being dirtied (on a per-transaction or per-second basis, for example).	
db block gets	8	Number of times a `CURRENT` block was requested **See Also:** "consistent gets"	
db block gets direct	8	Number of times a `CURRENT` block was requested bypassing the buffer cache (for example, a direct load operation). This is a subset of "db block gets" statistics value.	
db block gets from cache	8	Number of times a `CURRENT` block was requested from the buffer cache. This is a subset of "db block gets" statistics value.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
DBWR buffers scanned	8	Total number of dirty and clean buffers Oracle looks at when scanning LRU sets for dirty buffers to clean. Divide by "DBWR lru scans" to find the average number of buffers scanned.	
DBWR checkpoint buffers written	8	Number of buffers that were written for checkpoints	
DBWR checkpoints	8	Number of times the DBWR was asked to scan the cache and write all blocks marked for a checkpoint or the end of recovery. This statistic is always larger than "background checkpoints completed".	
DBWR cross instance writes	40	Real Application Clusters only: Number of blocks written to satisfy a lock request from another instance	
DBWR free buffers found	8	Number of clean buffers that DBWR found when it was requested to make free buffers. Divide by "DBWR make free requests" to find the average number of reusable buffers at the end of each LRU.	
DBWR lru scans	8	Number of times that DBWR scans the LRU queue looking for buffers to write. This count includes scans to fill a batch being written for another purpose (such as a checkpoint). This statistic is always greater than or equal to "DBWR make free requests".	
DBWR make free requests	8	Number of requests to DBWR to make some free buffers for the LRU	
DBWR revisited being-written buffer	8	Number of times that DBWR tried to save a buffer for writing and found that it was already in the write batch. This statistic measures the amount of "useless" work that DBWR had to do in trying to fill the batch. Many sources contribute to a write batch. If the same buffer from different sources is considered for adding to the write batch, then all but the first attempt will be "useless" because the buffer is already marked as being written.	
DBWR summed scan depth	8	The current **scan depth** (number of buffers examined by DBWR) is added to this statistic every time DBWR scans the LRU for dirty buffers. Divide by "DBWR lru scans" to find the average scan depth.	
DBWR transaction table writes	8	Number of rollback segment headers written by DBWR. This statistic indicates how many "hot" buffers were written, causing a user process to wait while the write completed.	
DBWR undo block writes	8	Number of rollback segment blocks written by DBWR	
DDL statements parallelized	32	Number of DDL statements that were executed in parallel	
deferred (CURRENT) block cleanout applications	128	Number of times cleanout records are deferred, piggyback with changes, always current get	
DFO trees parallelized	32	Number of times a serial execution plan was converted to a parallel plan	
dirty buffers inspected	8	Number of dirty buffers found by the user process while the it is looking for a buffer to reuse	
DML statements parallelized	32	Number of DML statements that were executed in parallel	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
enqueue conversions	4	Total number of conversions of the state of table or row lock	
enqueue deadlocks	4	Total number of deadlocks between table or row locks in different sessions	
enqueue releases	4	Total number of table or row locks released	
enqueue requests	4	Total number of table or row locks acquired	
enqueue timeouts	4	Total number of table and row locks (acquired and converted) that timed out before they could complete	
enqueue waits	4	Total number of waits that occurred during an enqueue convert or get because the enqueue get was deferred	
exchange deadlocks	8	Number of times that a process detected a potential deadlock when exchanging two buffers and raised an internal, restartable error. Index scans are the only operations that perform exchanges.	
execute count	64	Total number of calls (user and recursive) that executed SQL statements	
free buffer inspected	8	Number of buffers skipped over from the end of an LRU queue in order to find a reusable buffer. The difference between this statistic and "dirty buffers inspected" is the number of buffers that could not be used because they had a user, a waiter, or were being read or written, or because they were busy or needed to be written after rapid aging out.	
free buffer requested	8	Number of times a reusable buffer or a free buffer was requested to create or load a block	
global cache blocks corrupt	40	Real Application Clusters only: Number of blocks that encountered a corruption or checksum failure during interconnect	
global cache convert time	40	Real Application Clusters only: Total time elapsed during lock converts	
global cache convert timeouts	40	Number of times lock converts in the global cache timed out	
global cache converts	40	Number of lock converts in the global cache	
global cache cr block log flushes	40	Number of log flushes of the consistent-read block	
global cache cr block log flush time	40	Total time spent by the BSP process in log flushes after sending a constructed consistent-read (CR) block. This statistic divided by "global cache cr blocks served" = log flush time per CR block.	
global cache cr block receive time	40	Total amount of time foreground processes waited for a CR block to be sent through the interconnect. This statistic divided by "global cache cr blocks received" = time waited per block.	
global cache cr block send time	40	Total time spent by the BSP process in sending constructed consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = send time per CR block.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
global cache cr block serve time	40	Total amount of time the BSP process took to construct consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = construction time per CR block.	
global cache cr blocks received	40	Total number of blocks received	
global cache cr blocks served	40	Total number of blocks constructed by the BSP process	
global cache cr requests blocked	40	Number of times foreground attempt to request a cr block and failed	
global cache cr timeouts	40	Number of times a foreground process requested a consistent-read (CR) block when the request timed out	
global cache defers	40	Number of times a lock was requested and the holder of the lock deferred the release	
global cache freelist waits	40	System configured with fewer lock elements than buffers. Number of times foreground has to wait for a lock element.	
global cache get time	40	Total time spent waiting. This divided by global cache gets = time waited per request.	
global cache gets	40	Number of locks acquired	
global cache prepare failures	40	Number of times a failure occurred during preparation for interconnect transfer	
global lock async converts	32	Total number of asynchronous global lock converts	
global lock async gets	32	Total number of asynchronous global lock gets	
global lock convert time	32	Total elapsed time in 10s of milliseconds of all synchronous and asynchronous global lock converts	
global lock get time	32	Total elapsed time in 10s of milliseconds of all synchronous and asynchronous global lock gets	
global lock releases	32	Total number of synchronous global lock releases	
global lock sync converts	32	Total number of synchronous global lock converts	
global lock sync gets	32	Total number of synchronous global lock gets	
hot buffers moved to head of LRU	8	When a hot buffer reaches the tail of its replacement list, Oracle moves it back to the head of the list to keep it from being reused. This statistic counts such moves.	
immediate (CR) block cleanout applications	128	Number of times cleanout records are applied immediately during consistent-read requests	
immediate (CURRENT) block cleanout applications	128	Number of times cleanout records are applied immediately during current gets. Compare this statistic with "deferred (CURRENT) block cleanout applications"	
index fast full scans (direct read)	64	Number of fast full scans initiated using direct read	
index fast full scans (full)	64	Number of fast full scans initiated for full segments	
index fast full scans (rowid ranges)	64	Number of fast full scans initiated with rowid endpoints specified	
instance recovery database freeze count	32	Number of times the database is frozen during instance recovery	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
kcmccs called get current scn	32	Number of times the kernel got the CURRENT SCN when there was a need to casually confirm the SCN	
kcmgss read scn without going to DLM	32	Number of times the kernel got a snapshot SCN without going to the distributed lock manager (DLM)	
kcmgss waited for batching	32	Number of times a database process is blocked waiting for a snapshot SCN	
leaf node splits	128	Number of times an index leaf node was split because of the insertion of an additional value	
lob reads	8	Number of LOB API read operations performed in the session/system. A single LOB API read may correspond to multiple physical/logical disk block reads.	
lob writes	8	Number of LOB API write operations performed in the session/system. A single LOB API write may correspond to multiple physical/logical disk block writes.	
lob writes unaligned	8	Number of LOB API write operations whose start offset or buffer size is not aligned to the internal chunk size of the LOB. Writes aligned to chunk boundaries are the most efficient write operations. The internal chunk size of a LOB is available through the LOB API (for example, DBMS_LOB.GETCHUNKSIZE()).	
logons cumulative	1	Total number of logons since the instance started. Useful only in V$SYSSTAT. It gives an instance overview of all processes that logged on.	
logons current	1	Total number of current logons. Useful only in V$SYSSTAT.	
messages received	128	Number of messages sent and received between background processes	
messages sent	128	Number of messages sent and received between background processes	
native hash arithmetic execute	64	Number of hash operations performed using native arithmetic rather than Oracle NUMBERs	
native hash arithmetic fail	64	Number of has operations performed using native arithmetic that failed, requiring the hash operation to be performed with Oracle NUMBERs	
next scns gotten without going to DLM	32	Number of system change numbers obtained without going to the distributed lock manager or server	
no buffer to keep pinned count	72	Number of times a visit to a buffer attempted, but the buffer was not found where expected. Like "buffer is not pinned count" and "buffer is pinned count", this statistic is useful only for internal debugging purposes.	
no work - consistent read gets	128	Number consistent gets that require neither block cleanouts nor rollbacks. **See Also:** "consistent gets"	
opened cursors cumulative	1	In V$SYSSTAT: Total number of cursors opened since the instance started. In V$SESSTAT: Total number of cursors opened since the start of the session.	
opened cursors current	1	Total number of current open cursors	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
opens of replaced files	8	Total number of files that had to be reopened because they were no longer in the process file cache	
opens requiring cache replacement	8	Total number of file opens that caused a current file in the process file cache to be closed	
OS All other sleep time	16	Time spent sleeping for reasons other than misses in the data segment (see "OS Data page fault sleep time"), kernel page faults (see "OS Kernel page fault sleep time"), misses in the text segment (see "OS Text page fault sleep time"), or waiting for an OS locking object (see "OS User lock wait sleep time"). An example of such a reason is expiration of quanta.	
OS Chars read and written	16	Number of bytes read and written	
OS Data page fault sleep time	16	Time spent sleeping due to misses in the data segment	
OS Input blocks	16	Number of read I/Os	
OS Involuntary context switches	16	Number of context switches that were enforced by the operating system	
OS Kernel page fault sleep time	16	Time spent sleeping due to OS kernel page faults	
OS Major page faults	16	Number of page faults that resulted in I/O	
OS Messages received	16	Number of messages received	
OS Messages sent	16	Number of messages sent	
OS Minor page faults	16	Number of page faults that did not result in an actual I/O	
OS Other system trap CPU time	16	Total amount of time to process system traps (as distinct from system calls)	
OS Output blocks	16	Number of write I/Os	
OS Process heap size	16	Size of area in memory allocated by the process. Typically this represents memory obtained by way of malloc().	
OS Process stack size	16	Size of the process stack segment	
OS Signals received	16	Number of signals received	
OS Swaps	16	Number of swap pages	
OS System call CPU time	16	Total amount of time spent executing in system mode	
OS System calls	16	Number of system calls	
OS Text page fault sleep time	16	Time spent sleeping due to misses in the text segment	
OS User level CPU time	16	Total amount of time spent executing in user mode	
OS User lock wait sleep time	16	Total amount of time sleeping while waiting for an OS locking object	
OS Voluntary context switches	16	Number of voluntary context switches (for example, when a process gives up the CPU by a SLEEP() system call)	
OS Wait-cpu (latency) time	16	Time spent sleeping while waiting for a CPU to become available	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
Parallel operations downgraded 1 to 25 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 25 to 50 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 50 to 75 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded 75 to 99 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers	
Parallel operations downgraded to serial	32	Number of times parallel execution was requested but execution was serial because of insufficient parallel execution servers	
Parallel operations not downgraded	32	Number of times parallel execution was executed at the requested degree of parallelism	
parse count (hard)	64	Total number of parse calls (real parses). A hard parse is a very expensive operation in terms of memory use, because it requires Oracle to allocate a workheap and other memory structures and then build a parse tree.	
parse count (total)	64	Total number of parse calls (hard and soft). A soft parse is a check on an object already in the shared pool, to verify that the permissions on the underlying object have not changed.	
parse time cpu	64	Total CPU time used for parsing (hard and soft) in 10s of milliseconds	3
parse time elapsed	64	Total elapsed time for parsing, in 10s of milliseconds. Subtract "parse time cpu" from the this statistic to determine the total waiting time for parse resources.	3
physical read bytes	8	Total size in bytes of all disk reads by application activity (and not other instance activity) only.	
physical read IO requests	8	Number of read requests for application activity (mainly buffer cache and direct load operation) which read one or more database blocks per request. This is a subset of "physical read total IO requests" statistic.	
physical read total bytes	8	Total size in bytes of disk reads by all database instance activity including application reads, backup and recovery, and other utilities. The difference between this value and "physical read bytes" gives the total read size in bytes by non-application workload.	
physical read total IO requests	8	Number of read requests which read one or more database blocks for all instance activity including application, backup and recovery, and other utilities. The difference between this value and "physical read total multi block requests" gives the total number of single block read requests.	
physical read total multi block requests	8	Total number of Oracle instance read requests which read in two or more database blocks per request for all instance activity including application, backup and recovery, and other utilities.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
physical reads	8	Total number of data blocks read from disk. This value can be greater than the value of "physical reads direct" plus "physical reads cache" as reads into process private buffers also included in this statistic.	
physical reads cache	8	Total number of data blocks read from disk into the buffer cache. This is a subset of "physical reads" statistic.	
physical reads direct	8	Number of reads directly from disk, bypassing the buffer cache. For example, in high bandwidth, data-intensive operations such as parallel query, reads of disk blocks bypass the buffer cache to maximize transfer rates and to prevent the premature aging of shared data blocks resident in the buffer cache.	
physical reads prefetch warmup	8	Number of data blocks that were read from the disk during the automatic prewarming of the buffer cache.	
physical write bytes	8	Total size in bytes of all disk writes from the database application activity (and not other kinds of instance activity).	
physical write IO requests	8	Number of write requests for application activity (mainly buffer cache and direct load operation) which wrote one or more database blocks per request.	
physical write total bytes	8	Total size in bytes of all disk writes for the database instance including application activity, backup and recovery, and other utilities. The difference between this value and "physical write bytes" gives the total write size in bytes by non-application workload.	
physical write total IO requests	8	Number of write requests which wrote one or more database blocks from all instance activity including application activity, backup and recovery, and other utilities. The difference between this stat and "physical write total multi block requests" gives the number of single block write requests.	
physical write total multi block requests	8	Total number of Oracle instance write requests which wrote two or more blocks per request to the disk for all instance activity including application activity, recovery and backup, and other utilities.	
physical writes	8	Total number of data blocks written to disk. This statistics value equals the sum of "physical writes direct" and "physical writes from cache" values.	
physical writes direct	8	Number of writes directly to disk, bypassing the buffer cache (as in a direct load operation)	
physical writes from cache	8	Total number of data blocks written to disk from the buffer cache. This is a subset of "physical writes" statistic.	
physical writes non checkpoint	8	Number of times a buffer is written for reasons other than advancement of the checkpoint. Used as a metric for determining the I/O overhead imposed by setting the `FAST_START_IO_TARGET` parameter to limit recovery I/Os. (Note that `FAST_START_IO_TARGET` is a deprecated parameter.) Essentially this statistic measures the number of writes that would have occurred had there been no checkpointing. Subtracting this value from "physical writes" gives the extra I/O for checkpointing.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
pinned buffers inspected	8	Number of times a user process, when scanning the tail of the replacement list looking for a buffer to reuse, encountered a cold buffer that was pinned or had a waiter that was about to pin it. This occurrence is uncommon, because a cold buffer should not be pinned very often.	
prefetched blocks	8	Number of contiguous and noncontiguous blocks that were prefetched	
prefetched blocks aged out before use	8	Number of contiguous and noncontiguous blocks that were prefetched but aged out before use	
process last non-idle time	128	The last time this process executed	3
PX local messages recv'd	32	Number of local messages received for parallel execution within the instance local to the current session	
PX local messages sent	32	Number of local messages sent for parallel execution within the instance local to the current session	
PX remote messages recv'd	32	Number of remote messages received for parallel execution within the instance local to the current session	
PX remote messages sent	32	Number of remote messages sent for parallel execution within the instance local to the current session	
queries parallelized	32	Number of SELECT statements executed in parallel	
recovery array read time	8	Elapsed time of I/O during recovery	
recovery array reads	8	Number of reads performed during recovery	
recovery blocks read	8	Number of blocks read during recovery	
recovery blocks read for lost write detection	8	Number of blocks read for lost write checks during recovery.	
recovery blocks skipped lost write checks	8	Number of Block Read Records that skipped the lost write check during recovery.	
recursive calls	1	Number of recursive calls generated at both the user and system level. Oracle maintains tables used for internal processing. When Oracle needs to make a change to these tables, it internally generates an internal SQL statement, which in turn generates a recursive call.	
recursive cpu usage	1	Total CPU time used by non-user calls (recursive calls). Subtract this value from "CPU used by this session" to determine how much CPU time was used by the user calls.	
redo blocks checksummed by FG (exclusive)	2	Number of exclusive redo blocks that were checksummed by the generating foreground processes. An exclusive redo block is the one whose entire redo content belongs to a single redo entry.	
redo blocks checksummed by LGWR	2	Number of redo blocks that were checksummed by the LGWR.	
redo blocks written	2	Total number of redo blocks written. This statistic divided by "redo writes" equals number of blocks per write.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
redo buffer allocation retries	2	Total number of retries necessary to allocate space in the redo buffer. Retries are needed either because the redo writer has fallen behind or because an event such as a log switch is occurring.	
redo entries	2	Number of times a redo entry is copied into the redo log buffer	
redo entries for lost write detection	2	Number of times a Block Read Record is copied into the log buffer.	
redo log space requests	2	Number of times the active log file is full and Oracle must wait for disk space to be allocated for the redo log entries. Such space is created by performing a log switch. Log files that are small in relation to the size of the SGA or the commit rate of the work load can cause problems. When the log switch occurs, Oracle must ensure that all committed dirty buffers are written to disk before switching to a new log file. If you have a large SGA full of dirty buffers and small redo log files, a log switch must wait for DBWR to write dirty buffers to disk before continuing. Also examine the **log file space** and **log file space switch** wait events in `V$SESSION_WAIT`	
redo log space wait time	2	Total elapsed waiting time for "redo log space requests" in 10s of milliseconds	3
redo ordering marks	2	Number of times that a system change number was allocated to force a redo record to have a higher SCN than a record generated in another thread using the same block	
redo size	2	Total amount of redo generated in bytes	
redo size for lost write detection	2	Total amount of Block Read Records generated in bytes.	
redo synch time	8	Elapsed time of all "redo synch writes" calls in 10s of milliseconds	3
redo synch writes	8	Number of times a change being applied to the log buffer must be written out to disk due to a commit. The log buffer is a circular buffer that LGWR periodically flushes. Usually, redo that is generated and copied into the log buffer need not be flushed out to disk immediately.	
redo wastage	2	Number of bytes wasted because redo blocks needed to be written before they are completely full. Early writing may be needed to commit transactions, to be able to write a database buffer, or to switch logs.	
redo write time	2	Total elapsed time of the write from the redo log buffer to the current redo log file in microseconds	3
redo writer latching time	2	Elapsed time in 10s of milliseconds needed by LGWR to obtain and release each copy latch	3
redo writes	2	Total number of writes by LGWR to the redo log files. "redo blocks written" divided by this statistic equals the number of blocks per write	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
remote instance undo block writes	40	Number of times this instance wrote a rollback segment so that another instance could read it	
remote instance undo header writes	40	Number of times this instance wrote a undo header block so that another instance could read it	
rollback changes - undo records applied	128	Number of undo records applied to user-requested rollback changes (not consistent-read rollbacks)	
rollbacks only - consistent read gets	128	Number of consistent gets that require only block rollbacks, no block cleanouts. **See Also:** "consistent gets"	
rows fetched via callback	64	Rows fetched via callback. Useful primarily for internal debugging purposes.	
serializable aborts	1	Number of times a SQL statement in a serializable isolation level had to abort	
session connect time	1	The connect time for the session in 10s of milliseconds. This value is useful only in `V$SESSTAT`. It is the wall clock time since the logon to this session occurred.	3
session cursor cache count	64	Total number of cursors cached. This statistic is incremented only if `SESSION_CACHED_CURSORS` > 0. This statistic is the most useful in `V$SESSTAT`. If the value for this statistic in `V$SESSTAT` is close to the setting of the `SESSION_CACHED_CURSORS` parameter, the value of the parameter should be increased.	
session cursor cache hits	64	Number of hits in the session cursor cache. A hit means that the SQL statement did not have to be reparsed. Subtract this statistic from "parse count (total)" to determine the real number of parses that occurred.	
session logical reads	1	The sum of "db block gets" plus "consistent gets". This includes logical reads of database blocks from either the buffer cache or process private memory.	
session pga memory	1	Current PGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.	
session pga memory max	1	Peak PGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.	
session stored procedure space	1	Amount of memory this session is using for stored procedures	
session uga memory	1	Current UGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.	
session uga memory max	1	Peak UGA size for a session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.	
shared io pool buffer get failure	128	Number of unsuccessful buffer gets from the shared I/O pool from instance startup time.	
shared io pool buffer get success	128	Number of successful buffer gets from the shared I/O pool from instance startup time.	
sorts (disk)	64	Number of sort operations that required at least one disk write Sorts that require I/O to disk are quite resource intensive. Try increasing the size of the initialization parameter `SORT_AREA_SIZE`. For more information, see "SORT_AREA_SIZE" on page 1-152.	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
sorts (memory)	64	Number of sort operations that were performed completely in memory and did not require any disk writes You cannot do much better than memory sorts, except maybe no sorts at all. Sorting is usually caused by selection criteria specifications within table join SQL operations.	
sorts (rows)	64	Total number of rows sorted	
SQL*Net roundtrips to/from client	1	Total number of Oracle Net Services messages sent to and received from the client	
SQL*Net roundtrips to/from dblink	1	Total number of Oracle Net Services messages sent over and received from a database link	
summed dirty queue length	8	The sum of the dirty LRU queue length after every write request. Divide by **write requests** to get the average queue length after write completion.	
switch current to new buffer	8	Number of times the CURRENT block moved to a different buffer, leaving a CR block in the original buffer	
table fetch by rowid	64	Number of rows that are fetched using a ROWID (usually recovered from an index) This occurrence of table scans usually indicates either non-optimal queries or tables without indexes. Therefore, this statistic should increase as you optimize queries and provide indexes in the application.	
table fetch continued row	64	Number of times a chained or migrated row is encountered during a fetch Retrieving rows that span more than one block increases the logical I/O by a factor that corresponds to the number of blocks than need to be accessed. Exporting and re-importing may eliminate this problem. Evaluate the settings for the storage parameters PCTFREE and PCTUSED. This problem cannot be fixed if rows are larger than database blocks (for example, if the LONG datatype is used and the rows are extremely large).	
table scan blocks gotten	64	During scanning operations, each row is retrieved sequentially by Oracle. This statistic counts the number of blocks encountered during the scan. This statistic tells you the number of database blocks that you had to get from the buffer cache for the purpose of scanning. Compare this value with the value of "consistent gets" to determine how much of the consistent read activity can be attributed to scanning.	
table scan rows gotten	64	Number of rows that are processed during scanning operations	
table scans (cache partitions)	64	Number of range scans performed on tables that have the CACHE option enabled	
table scans (direct read)	64	Number of table scans performed with direct read (bypassing the buffer cache)	
table scans (long tables)	64	Long (or conversely short) tables can be defined as tables that do not meet the short table criteria as described in table scans (short tables)	

Table E–1 (Cont.) Database Statistics Descriptions

Name	Class	Description	TIMED_ STATISTICS
table scans (rowid ranges)	64	During parallel query, the number of table scans conducted with specified ROWID ranges	
table scans (short tables)	64	Long (or conversely short) tables can be defined by optimizer hints coming down into the row source access layer of Oracle. The table must have the CACHE option set.	
total file opens	8	Total number of file opens performed by the instance. Each process needs a number of files (control file, log file, database file) in order to work against the database.	
transaction lock background get time	128	Useful only for internal debugging purposes	
transaction lock background gets	128	Useful only for internal debugging purposes	
transaction lock foreground requests	128	Useful only for internal debugging purposes	
transaction lock foreground wait time	128	Useful only for internal debugging purposes	
transaction rollbacks	128	Number of transactions being successfully rolled back	
transaction tables consistent read rollbacks	128	Number of times rollback segment headers are rolled back to create consistent read blocks	
transaction tables consistent reads - undo records applied	128	Number of undo records applied to transaction tables that have been rolled back for consistent read purposes	
Unnecessary process cleanup for SCN batching	32	Total number of times that the process cleanup was performed unnecessarily because the session or process did not get the next batched SCN. The next batched SCN went to another session instead.	
user calls	1	Number of user calls such as login, parse, fetch, or execute When determining activity, the ratio of user calls to RPI calls, give you an indication of how much internal work gets generated as a result of the type of requests the user is sending to Oracle.	
user commits	1	Number of user commits. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.	
user rollbacks	1	Number of times users manually issue the `ROLLBACK` statement or an error occurs during a user's transactions	
write clones created in background	8	Number of times a background or foreground process clones a `CURRENT` buffer that is being written. The clone becomes the new, accessible `CURRENT` buffer, leaving the original buffer (now the clone) to complete writing.	
write clones created in foreground	8	Number of times a background or foreground process clones a `CURRENT` buffer that is being written. The clone becomes the new, accessible `CURRENT` buffer, leaving the original buffer (now the clone) to complete writing.	

Index